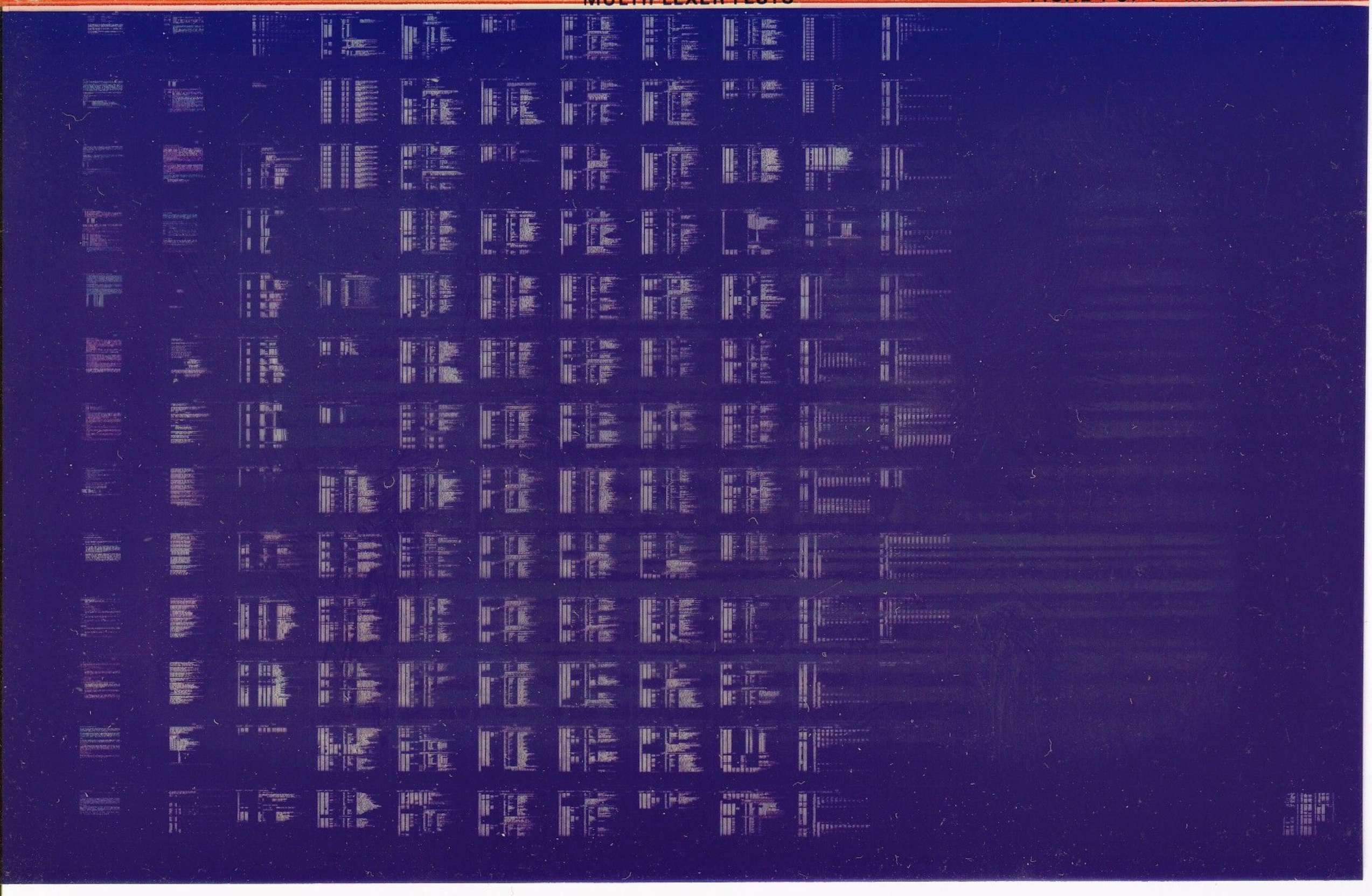
DZ11

8 LINE ASYNCHRONOUS
MD-11-DZDZA-E
MULTIPLEXER TESTS

EP-DZDZA-E-DL

FICHE 1 OF 1

MAR 1978
digital
MADE IN USA



PRODUCT CODE: MAINDEC-11-DZDZA-E-D

PRODUCT NAME: DZ11 8 LINE ASYNC MUX TESTS

DATE RELEASED: FEB 1978

MAINTAINER: DIAGNOSTIC ENGINEERING

1. ABSTRACT

The function of the DZ11 diagnostics is to verify the option operates according to specifications. The diagnostics also verify that the DZ11 operates in its environment such as the system in which it is installed.

Parameters may be supplied to the program by either 'AUTO SIZING' or input from the user on the console by having SW00=1 at start time. Auto sizing will be done only the first time the program is started and SW07=0 and SW00=0 and SW03=0. The AUTOSIZER is designed to detect DZ11 device addresses and vectors and to determine whether the DZ11 that is detected is an EIA or 20mA board. All remaining parameters default to certain values (see SEC.B.5). Console input may be controlled at any start time through the use of SW00, SW03, SW04, and SW06 (see SEC. 4.1.1 for a detailed description of these switches).

Currently there is one standalone diagnostic (DZDZA), one system module for DEC X/11 (DZAA), and an online overlay for DZITA (ITEP) - DZDZB. (ITEP) - DZDZB.

DZDZA will test all parts of the DZ11 such as cables, dist pnl., and the interface module itself.

2. REQUIREMENTS

2.1 EQUIPMENT

Any PDP11 family CPU (WITH MINIMUM 8K MEMORY)
ASR 33 (or equivalent for console)

DZ11	INTERFACE MODULE (M7819(EIA), M7814(20MA))
H3271	Staggered turnaround connector for EIA module.
H3190	Staggered turnaround connector for 20mA module.
H325	Cable turnaround and dist pnl testing for EIA module.
H315	This may be subsituted for H325.

NOTE: A staggered turnaround connector is needed in order to test the PARITY and BREAK logic.

2.2 STORAGE

Program will use all 8K of memory except where ABL and BOOTSTRAP LOADER reside. Location 1500 thru 2000 are especially to be noted and to be untouched by operator after parameters have been input from console (SW00=1); or after the 'AUTO SIZING' has been done. These locations may be changed if the user understands their meaning and different parameters are required.

3. LOADING PROCEEDURE

3.1 METHOD

All programs are in absolute format and are loaded using the ABSOLUTE LOADER. NOTE: if the diagnostics are on a media such as DISK ,MAGTAPE,DECTAPE, or CASSETTE; follow instructions for the monitor which has been provided on that specific media.

ABSOLUTE LOADER starting address *500

MEMORY	* SIZE
4k	17
8k	37
12k	57
16k	77
20k	117
24k	137
28k	157

3.1.1 Place address of ABS loader into switch register.
(also place 'HALT' SW up)

3.1.2 Depress 'LOAD ADDRESS' key on console and release.

3.1.3 Depress 'START KEY' on console and release (program should now be loading into CPU)

4. STARTING PROCEEDURE

A. Set switch register to 000200
B. Depress 'LOAD ADDRESS' key and release
C. Set SWR to zero for 'AUTO SIZING' or set SW00=1 for user parameter input from console terminal. On first start if SW07=1 and SW00=0 the program will default to console parameter input (SW00=1).
D. Depress 'START KEY' and release, the program will type Maindec Name and program name (if this was the first start up of the program or parameters were changed by SW00=1) and also the following:

```
'MAP OF DZ11 STATUS'
1500    160100
1502    000300
1504    000005
1506    000377
1510    017470
1512    000000
```

The above is only an example! This would indicate the status table starting at add. 1500 in the program. THE STATUS TABLE MUST BE VERIFIED BY THE USER IF AUTO SIZING IS DONE. For information of status table see section 8.4 for help.

The program will type "Running" and proceed to run the diagnostic.

4.1 CONTROL SWITCH SETTINGS

NOTE: If there is no real SWR (177570); SWR may be modified at Loc:176 or by hitting Control "G" (^G) on console terminal.

```
  SW 15   Set:  Halt on error
  SW 14   Set:  Loop on current test
  SW 13   Set:  Inhibit error print out
  SW 12   Set:  Inhibit **ALL** type out/bell on error.
  SW 11   Set:  Inhibit iterations.  (quick pass)
  SW 10   Set:  Escape to next test
  SW 09   Set:  Loop with current data
  SW 08   Set:  Catch error and loop on it
  SW 07   Set:  NO AUTO SIZE.  If 1st start of program after loading the
                operator must input address and vector from console.
  SW 06   Set:  Reselect DZ11's desired active
  SW 05   Set:  Reserved
  SW 04   Set:  Select delay parameter (see SEC. 4.1.1)
  SW 03   Set:  Extra parameter input  (see SEC. 4.1.1)
  SW 02   Set:  Lock on selected test
**SW 01   Set:  Restart program at selected test
 *SW 00   Set:  Get users parameters from console
```

* For Echo or Cable tests (program started at loc. 210) this switch set to 1 allows the user to type in the Vector and the CSR for the DZ11 under test.

** For Echo or Cable test this switch set to 1 allows the selection of either the Echo or Cable test, baud rate, and the line number under test.

4.1.1 SWITCH REGISTER CONTROL OF PARAMETER INPUT FROM CONSOLE

SW 00 GET USERS PARAMETERS FROM CONSOLE. Setting this switch at start up time allows the user to input at the Console terminal the following parameters: base device address, base vector address, bus request level, declare EIA or 20mA module, mode of operation (External,Internal, or Staggered), and the number of DZ11's that are running. Using this switch alone defaults the following parameters: all 8 lines are set to be tested on each DZ11, the default baud rate is set at 19.2 kbaud, and the character length for the majority of testing is set at eight bits per character with two stop bits.

SW 03 EXTRA PARAMETER INPUT. Setting this switch at start up time provides the user with the ability to set the lines active for testing and to set the default baud rate used for the majority of the diagnostic tests. The Delay Parameter is automatically adjusted to the baud rate given by the user.

SW 04 SELECT DELAY PARAMETER. The DELAY parameter this switch controls determines the length of time the program stalls waiting for a character to be completely transmitted or received. This delay count is automatically set to provide enough delay time for the default baud rate specified when running the program on an 11/45 with bipolar memory. When running this program on a faster processor the delay parameter should be adjusted proportionally higher than the following defaulted values:

```
2450        ;time for   50 baud
1560        ;time for   75 baud
1120        ;time for  110 baud
0750        ;time for  134 baud
0660        ;time for  150 baud
0330        ;time for  300 baud
0150        ;time for  600 baud
0060        ;time for 1200 baud
0040        ;time for 1800 baud
0030        ;time for 2000 baud
0020        ;time for 2400 baud
0010        ;time for 3600 baud
0001        ;time for 4800 baud
0001        ;time for 7200 baud
0001        ;time for 9600 baud
0001        ;time for 19.2 kbaud
```

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 06 RESELECT DZ11'S DESIRED ACTIVE. Please note that a message is typed out for setting the switch register equal to DZ11's active. This means if the system has four DZ11s; bits 00,01,02,03 will be set in loc 'DZACTV' from the switch register. Using this switch(SW06) alters that location;therefore if four DZ11s are in the system ***DO NOT*** set switches greater than SW 03 in the up position. This would be a fatal error. do not select more active DZ11s than has been given information about in parameter input (SW00=1)

METHOD: A: Load address 200
B: Start with SW 06=1
C: Program will type message
D: Set the BINARY number of DZ11s desired active EXAMPLE: 1=1 DZ11; 3=2 DZ11; 7=3 DZ11; 17=4 DZ11 37=5 DZ11 etc/aa PRESS CONTINUE.
E: Number (IF VALID) will be in data lights (excluding 11/05)
F: Set with any other switch settings desired. PRESS CONTINUE.

SW 01 RESTART PROGRAM AT SELECTED TEST it is strongly suggested that at least one pass has been made before trying to select a test that is not in the order of sequence the reason being is that the program has to clear areas and set up parameters. Note: if running multiple DZ11's; the DZ11 you desire to be under test must be selected by the use of SW06 before locking on the test. In other words; each time the program is started; the first DZ11 will be selected to be under test unless SW06 is used to select only one.

SW 09 LOOP ON CURRENT DATA: this switch will only work if call 'SCOPI' is in that test. The reason being that most tests deal with blocks of different data to be sent or received all at once thus in block data, one pattern can't be singled out.
This switch is designed to provide an aid for a trained troubleshooter to sample various signals on the module and is not meant to be used as a general user control switch.

SW 04 SELECT DELAY PARAMETER: THIS SWITCH SHOULD BE USED WITH CARE AS TOO SHORT A DELAY WILL CAUSE VALID TESTS TO FAIL ON CERTAIN PROCESSORS. IT IS RECOMMENDED THAT THIS SWITCH ONLY BE USED IN CONJUNCTION WITH SCOPE LOOPS, E.G, SW 14,9,4,1 SET; SW 9,4,2,1 SET. THE SHORTEST PARAMETER IS 1; THE LONGEST ACCEPTED IS 177776. (see SEC. 4.1.1)

4.1.3 SWITCH REGISTER PRIORITIES

ERROR SWITCHES

1. SW 12 Delete print out/bell on error.
2. SW 13 Delete error printout.
3. SW 15 Halt on the error.
4. SW 08 Goto beginning of the test(on error).
5. SW 10 Goto next test(on error).

SCOPE SWITCHES

1. SW 09 (if enabled by 'SCOP1'). If an '*' is printed in front of the test no. on an error report (ex. *TEST NO. 10) SW09 is incorporated in that test and therefore SW09 is *usually* the best switch for the scope loop (SW14=0, SW10=0, SW09=1, SW08=0) if the program user is technically trained to electronically isolate signal problems on the DZ11 module.
   If SW09 is not enabeled; and there is a *HARD* error (constant); SW08 is best.
2. For intermittent errors either start the program with SW01 and SW02 set which will allow the user to lock on a selected test, or else set SW14 as an error is being typed out on the terminal. SW14 will continue to loop on that test regardless of whether an error occurs.
3. SW 14 Loop on current test.

4.2 STARTING ADDRESS

SA 200 - Address 200 is for normal execution of the diagnostic. This will do the major testing necessary for verification of hardware.

SA 210 - CABLE/ECHO - Terminal Tests. Starting at address 210 will give the user the option to verify the EIA cables at the dist pnl or verify a true link to any DEC supported terminal supported by the DZ11.

NOTE: If address 000042 is non-zero the program assumes it is under ACT11 or XXDP control and will act accordingly. After *ALL* available DZ11's are tested the program will return to 'XXDP' or 'ACT-11'.

5. OPERATING PROCEDURE

When program is initially started messages as described in section four will be printed and program will begin running the diagnostic.

5.1 NORMAL START OF DIAGNOSTIC

On the first start of the diagnostic at address 200; if auto sizing is not used or whenever SW00=1; the following questions are asked and must be answered.

"1ST CSR ADDRESS (160000:163700): "

You must type in the first DZ11 CSR in the system you wish testing to begin at. RANGE: 160000:163700

"1ST VECTOR ADDRESS (300:770): "

You must type in the vector of the first DZ11 in the system under test. RANGE 300:770

"BR LEVEL (4:6): "

type in the priority level of the DZ11 that the above information has been given about. RANGE 4 or 5 or 6.

"TYPE "A" FOR EIA MODULE OR "B" FOR 20MA (A:B): "

Type "A" if running a DZ11-A,B,E (EIA).
Type "B" if running a DZ11-C,D,F (20MA).
Typing a <CR> defaults to EIA MODULES.

"MAINTENANCE MODE
[EXTERNAL <H325>-EIA ONLY (E)]
[INTERNAL <DZCSR03=1> (I)]
[STAGGERED <H3271>-EIA ONLY (S)]
[STAGGERED <H3190>-20mA ONLY (S)] :

Type "E" or "I" or "S" depending on which mode you wish to run in. If running "EXTERNAL"; all selected lines must be terminated by an H325 test connector.

"# OF DZ11'S <IN OCTAL> (1:20): "

Type total number of DZ11's to be tested in the system. RANGE is 1 thru 20 in octal.

********* IF SW03=1 THEN *********
If SW03=1 the following will be printed.

"LINES ACTIVE BY BIT <IN OCTAL> (001:377):"

Each bit represents a line and any combination of lines may be selected (HOWEVER IN STAGGERED MODE TWO ADJACENT LINES MUST BE SELECTED (0-1, 2-3, 4-5, 6-7))..

"DEFAULT BAUD RATE <IN OCTAL> (00:17): "

This gives the user a chance to change the default baud rate used in APP. 90% of the test. Baud rate choices are:
"00"(50 baud),"01"(75 baud),"02"(110 baud),"03"(134 baud),
"04"(150 baud),"05"(300 baud),"06"(600 baud),"07"(1200 baud),
"10"(1800 baud),"11"(2000 baud),"12"(2400 baud),"13"(3600 baud),
"14"(4800 baud),"15"(7200 baud),"16"(9600 baud),"17"(19.2 kbaud)
Low default baud rates are not suggested since they lengthen the time to complete a program pass dramatically.
****

It is important to note that all DZ11's in the system must be CONTIGIOUS for both ADDRESS and VECTORS. Also all the EXTRA PARAMETERS other than CSR and VECTORS are given to the EXISTING DZ11's in the system. If not all DZ11's are same priority or if the mode of operation is different for each DZ11; THIS MUST BE "PATCHED" INTO THE CORRECT STATUS MAP ENTRY which is printed at start time. An alternative is to put SW00=1 at start time; answer questions about DZ11 under test and INDICATE ONLY 1 DZ11 in the system. IF THE STATUS MAP IS TO BE "PATCHED" IT MUST BE DONE AFTER THE QUESTIONS ARE ANSWERED OR AFTER THE AUTO SIZE.

5.2 HOW TO RUN THE "CABLE/ECHO" TESTS.

Normal starting for the first time would be: LOAD ADDRESS 210; START WITH THE SWR EQUAL TO 003.
NOTE: SW00=1 ASKS FOR "VECTOR" AND "CSR"
SW01=1 ASKS FOR "WHICH TEST EC40 OR CABLE", "BAUD RATE", "LINE" UNDER TEST. Program will print ut:

"VECTOR ADDRESS-"

You type vector with a <CR>.

"CONTROL REGISTER ADDRESS-"

You type in DZCSR under test.

"WHICH TEST ? ECHO OR CABLE (E OR C)"

Lets do the CABLE TEST first. **THIS TEST IS ONLY TO BE DONE ON THE EIA VERSION OF THE DZ11 NOT THE 20MA VERSION". Type "C" <CR>

"BAUD RATE- "

type either 50, 110, 135, 150, 300, 600, 1200 1800, 2000, 2400, 3600, 4800, 7200, 9600 followed by <CR>

"LINE: "

You type the line which has the H325 test connector. (Type either 0, 1, 2, 3, 4, 5, 6, 7) Program will then print:

"CABLE TEST"

and if everything is working; the following will be printed:

"PASS DONE."
"PASS DONE."
etc.

to change lines; HIT ANY PRINTING KEY ON YOUR CONSOLE TERMINAL WHILE THE PROGRAM IS RUNNING and the following will be printed:

"LINE: "

Now change the H325 test connector to another line and type the new line. Program will then print:

"CABLE TEST"
"PASS DONE."
"PASS DONE."

Continue this operation until all lines are tested.

5.3 ECHO TEST

If program has already been started at 210 and the vector and address have been typed in; just load address 210 and start with SWR equal to 002. program will print:

"WHICH TEST ? ECHO OR CABLE (E OR C)"

Now type an "E" to do the ECHO TEST. program will print:

"BAUD RATE-"

Type BAUD RATE at which the terminal is set that is connected to the DZ11 dist pnl. Baud rate choices are: 50, 75, 110, 135, 150 300, 600 , 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600. The program will then print:

LINE: "

Type the line the terminal is connected to at the dist pnl then the program will print:

"TERMINAL ECHO TEST"

*** AT THIS POINT THE MESSAGE:

"THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 0123456789"

SHOULD BE PRINTED ON THE TERMINAL CONNECTED TO THE DZ11. IF THIS MESSAGE IS DESIRED TO BE CONTINIOUSLY OUTPUT; SET THE SWR TO 377 (SWR=377) WHILE IT IS BEING OUTPUT OR WHEN THE LINE NO. IS REQUESTED ABOVE. WHEN THIS MESSAGE IS DONE AND THE SWR IS NOT EQUAL TO 377; THE CONSOLE WILL PRINT:

"TYPE A CHAR. ON DZ11 TERMINAL"

any printable char hit on DZ11 terminal should be echoed back on the terminal. **IF YOU HIT CNTRL C <↑C> ON THE DZ11 TERMINAL THE PROGRAM WILL PRINT:

"PASS DONE."

on the CONSOLE terminal and the "QUICK BROWN FOX" will be printed on DZ11 terminal again and the echo test will be running. TO CHANGE LINES: type any printable character on the CONSOLE TERMINAL (not the DZ11 terminal). The program will again type "LINE: " and wait for a response.

5.4 PROGRAM AND/OR OPERATOR ACTION

The variety of program Control Switches provided in this Diagnostic Package is designed to provide the user with a wide range of trouble-shooting techniques. Before the user attempts to run this diagnostic he should become familiar with the use of these Control Switches and their restrictions. (See Sec. 4.1, 4.1.1, 4.1.2, 4.1.3)

When the program detects an error the TEST NUMBER and PC will be typed out and possibly an error message (depending on the particular error). If it is necessary to know more information concerning the error report then look in the program listing for that TEST NUMBER and then note the PC of the error report. The reason for the error report will become clearer when reading the comments in the program listing.

6. ERRORS

As described previously there will always be a TEST NUMBER and PC typed out at the time of an error (providing SW 13=0 and SW 12=0). In most cases additional information will be supplied to the the error message which is to give the operator an indication of the error.

6.2 ERROR RECOVERY

If for some reason the DZ11 should 'HANG THE BUS' (gain control of bus so that console manual functions are inhibited) an init or power down/up is necessary for operator to regain control of cpu. If this should happen; look in location 'TSTNO' (address 1216)for the number of the test that was running at the time of the catastrophic error. In this way the operator will have an idea as to what the DZ11 was doing at the time of the error.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

See section 4.1.2
Status table should be verified regardless of how program was started. Also it is important to use this listing along with the information printed on the TTY to completly isolate problems.

7.2 OPERATING RESTRICTIONS

Parameter must be input from user OR APT if "AUTO SIZING" is not used.

8. MISCELLANEOUS

8.1 EXECUTION TIME

All DZ11 device diagnostics will give an 'END PASS' message (providing no errors and sw12=0) within 2 min. This is assuming SW11=1 (INHIBIT ITERATIONS) is set to give the fastest possible execution. The actual execution time depends greatly on the PDP11 CPU configuration. An 11/40 with Core memory will take around 100 seconds to execute a pass with no iterations and about 400 seconds to execute a fully iterated pass. Any other PDP11 CPU type will execute a pass in time proportional to the execution speed of the CPU 's memory in relation to that of an 11/40.

8.2 PASS COMPLETE

NOTE: *EVERY* time the program is started; the tests will run as if SW11 (delete iterations) was up (=1). This is to 'VERIFY NO *HARD* ERRORS' as soon as possible. Therefore the first pass -EACH TIME PROGRAM IS STARTED- will be a 'QUICK PASS' until all DZ11's in system are tested. When the diagnostic has completed a pass the following is an example of the print out to be expected.

END PASS DZDZA-D CSR: 160010 VEC: 300 PASSES: 000001 ERRORS: 000000

NOTE: The numbers for CSR and VEC are not necessarily the values for the device. They are only for this example

8.4 KEY LOCATIONS

$LPADR (1126)	Contains the address where program will return when iteration count is reached or if loop on test is asserted.
NEXT (1360)	Contains the address of the next test to be performed.
$TSTNM (1122)	Contains the number of the test now being performed.
RUN (1406)	The bit in 'RUN' always points one past the DZ11 currently being tested. EXAMPLE: (RUN) 1304/0000000001000000 Means that DZ11 no.05 is the DZ11 now running.
STATUS MAP (1500)-(2000)	These locations contain the information needed to test up to 16 (decimal) DZ11s sequentialy. they contain the CSR,VECTOR and STATUS concerning the configuration of each DZ11.
DZACTV (1404)	Each bit set in this location indicates that the associated DZ11 will be tested in turn. EXAMPLE: (DZACTV) 1300/0000000000011111 means that DZ11 no. 00,01,02,03,04 will be tested. EXAMPLE: (DZACTV) 1300/0000000000010001 Means that DZ11 no. 00,04 will be tested.
$BASE (1310)	Contains the receiver CSR of the current DZ11 under test.

8.4A MORE ON THAT 'STATUS TABLE' (1500-2000)

```
        'MAP OF DZ11 STATUS'
1500    160100
1502    000300
1504    000005
1506    000377
1510    017470
1512    000000
```

The above information will be repeated for each of up to 16 DZ11's in the system(these will follow under this table). EXPLANATION:

1500 160100 This is the system control register for the 1st DZ11 in the system.

1502 000300 This is vector 'A' for the first DZ11 in the system.

1504 000005 This represents the bus interrupt priority level of the DZ11. BIT15 of this location indicates either EIA or 20MA. If BIT15=0 module should be an M7819, if bit15=1 module should be an M7814.

1506 000377 This is the binary representation of what lines are to be tested.

1510 017470 This is the parameter location used in most of the tests. It indicates parameters of: RX ON, SPEED SELECT 17 (19.2K BAUD) EIGHT BITS PER CHAR, AND TWO STOP BITS. The user may alter the stop bits and the speed, but the remaining parameters should be left alone.
This location is used to load the DZ11 Line Parameter Register for each line. The meaning of the bits set in this location is the same as the function of the related bits in the device Line Parameter Register.

1512 000000 This location will contain either all zeros indicating that internal loop was selected as mode of operation or it will contain 10000 indicating that "staggered mode" was selected or it will contain 000200 indicating that "external" was the mode selected.

The above is repeated for each DZ11 in the system. The table is filled by AUTO SIZING or by the manual parameter input program as described previously. Also if desired by user; the locations may be altered by hand (toggled in) to suit the specific configuration.

8.5 *** METHOD OF AUTO SIZING ***

8.5.1 FINDING THE CONTROL STATUS REGISTER.

The program will start at address 160000 and start 'REFERENCING' the address in the pointer. If a NON-EX MEMORY TRAP occures, the pointer (holding 160000) is updated by 10 and the above is repeated until address 163700 is reached. If a 'SLAVE SYNC RESPONSE' was issued by the DZ11 (or any other device) (no nxm trap), "MASTER SCAN ENABLE" is attempted to be set and the "TCR" bit for line 7 is set. "TRDY" is then tested to be set and both "TCR07" AND "MASTER SCAN ENABLE" are tested to be still set. If all of this worked; then a "DEVICE CLEAR" is issued testing that the bit can be read back and that after some time it self clears. If all of the above worked; this device is assumed to be a DZ11. If any of the above failed; updating of the pointer is done and the sequence is repeated.
NOTE: If the program does not find your DZ11; something is wrong and AUTO SIZING should not be done.
After identifying a DZ11 the program then attempts to set all DTR bits in Device Register 4. If any DTR bits did set the module is assumed to be an EIA module (M7819) otherwise the status map entry is set for 20mA (M7814).

8.5.2 FINDING THE VECTOR

The vector area (address 300-776) is filled with the instruction IOT and '.+2' (next address). Bit14 and Bit5 (TX INTERUPT ENABLE AND MSTSCAN ENABLE) are set into the DZCSR. "TCR07" is then set. a delay is made and if no interupt occures (because of a bad DZ11) the program assumes vector address 300 and the problem should be fixed in the diagnostic. Once the problem is fixed; the program should be re-setup again to get correct vector. If an interupt occurred; the address to which the DZ11 interupted to is picked up and reported as the vector. NOTE: if the vector reported is not the vector set up by you; there is a problem and AUTO SIZING should not be done.

8.5.3 PARAMETER ASSUMPTIONS.

Since too much hardware would need to be turned on to SIZE the rest of the parameters; the program must assume the remaining variations. The result if not to your specific configuration may be altered by hand (toggle in) if desired. In this way 95% of the parameter setup was done by the program and 5% by you.
THEREFORE:
1) BUS PRIORITY IS SET TO LEVEL5.
2) ALL EIGHT LINES ARE ASSUMED TO BE TESTED.
3) DEFAULT BAUD RATE IS SET TO 17 (19.2 K).
4) MODE OF OPERATION IS "INTERNAL MODE".

For all parameter adjustments please refer to section 8.4a for greater detail.

9.0 RUNNING THE DZ11 DIAGNOSTIC UNDER APT

9.1.1 THE APT INTERFACE

DZDZA has been redesigned to be compatible with the APT-Automated Product Test system. It can be run as a standalone diagnostic or in either of the APT modes. Certain variables in the original APT module were reassigned to the areas set aside for APT interfacing. These new variables generally begin with a dollar sign ($), e.g., $DEVM, $BASE.

9.1.2 SETTING UP THE DIAGNOSTIC USING APT

The diagnostic uses several variables in the region subtitled 'APT Mailbox-Etable'. These variables are:

$SWREG - used if a software switch register is desired while under apt .

$VECT1 - used to specify the interrupt level and the first vector address

$BASE - used to indicate bottom address of DZ11 under test

$DEVM - a bit map representing which DZ11's will be tested

$CDW1 - used to indicate which lines to run on all DZ11's

$DDW0 - each of the $DDW words describes the parameters (LPR) for a particular DZ11, going up to 16 DZ11's

9.1.3 RUNNING UNDER APT

The user should be familiar with the APT system. The APT timing parameters for the DZ11 diagnostic were based on an 11/40 processor. It may be necessary to add a few more seconds if the diagnostic is out on an 11/05 processor.

All of the variables mentioned in section 9.1.2 should be set up prior to running the diagnostic under APT.

NOTE

Be sure $BASE points to the first DZ11 before running

Based on these values, the diagnostic will set up the status table. The user is then free to monitor under APT as normal.

DOCUMENT
****
DZDZAE LST
****

```
        4       COPYRIGHT (C) 1977
                DIGITAL EQUIPMENT CORP.
                MAYNARD, MASS. 01754

                THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.

       24       INITIAL ADDRESS OF THE STACK POINTER *** 1120 ***

       c9       MISCELLANEOUS DEFINITIONS

       41       GENERAL PURPOSE REGISTER DEFINITIONS

       53       PRIORITY LEVEL DEFINITIONS

       63       "SWITCH REGISTER" SWITCH DEFINITIONS

       91       DATA BIT DEFINITIONS (BIT00 TO BIT15)

      119       BASIC "CPU" TRAP VECTOR ADDRESSES

      355       THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                USED IN THE PROGRAM.

      425                               BITS 15-11=CPU TYPE
                                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                              11/70=06,PDQ=07,Q=10
                                        BIT 10=REAL TIME CLOCK
                                        BIT  9=FLOATING POINT PROCESSOR
                                        BIT  8=MEMORY MANAGEMENT

      433                               MEM.TYPE BYTE   --  (HIGH BYTE)
                                              900 NSEC CORE=001
                                              300 NSEC BIPOLAR=002
                                              500 NSEC MOS=003

      438                                     MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE

      476       THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                NOTE1:  IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                NOTE2:  EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

      482               EM                ;;POINTS TO THE ERROR MESSAGE
                        DH                ;;POINTS TO THE DATA HEADER
                        DT                ;;POINTS TO THE DATA
                        DF                ;;POINTS TO THE DATA FORMAT
```

```
1090    INCREMENT THE PASS NUMBER ($PASS)
        IF THERES A MONITOR GO TO IT
        IF THERE ISN'T JUMP TO CYCLE

1151    THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
        AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
        AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
        THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
        SW14=1  LOOP ON TEST
        SW11=1  INHIBIT ITERATIONS
        CALL
                SCOPE           ;;SCOPE=IOT

1227    ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
        THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
        NOTE1:  $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
        NOTE2:  $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
        NOTE3:  $FILLC CONTAINS THE CHARACTER TO FILL AFTER.

        CALL:
        1) USING A TRAP INSTRUCTION
                TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
        OR
                TYPE
                MESADR


1932    ROUTINE USED TO "AUTO SIZE" THE DZ11
        CSR AND VECTOR.
        NOTE:   THE CSR MAY BE ANY WHERE IN THE FLOATING
                ADDRESS RANGE (160000:163710)
                AND THE VECTOR MAY BE ANY WHERE IN THE
                FLOATING VECTOR RANGE (300:770)


2054    ********************** TEST 1 ********************************
        THIS TEST PROVES THE SLAVE SYNC RESPONSE
        DURING A READ OR WRITE TO THE FOLLOWING ADDRESS:
                DZCSR, DZRBUF, DZTCR, DZMSR

2097    ********************** TEST 2 ********************************
        THIS TEST PROVES THAT BIT "DCLR"
        CAN BE SET AND THAT IT WILL CLEAR
        BY ITSELF AFTER A PERIOD OF TIME.

2127    ********************** TEST 3 ********************************
        TEST TO VERIFY THAT BIT "MAINT" CAN
        BE SET. THEN VERIFY THAT BIT "MAINT" CAN
        BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
        VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
        CLEARED BY A "DEVICE CLEAR"
```

```
2159    ********************** TEST 4 ***********************
        TEST TO VERIFY THAT BIT "MSENAB" CAN
        BE SET. THEN VERIFY THAT BIT "MSENAB" CAN
        BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
        VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
        CLEARED BY A "DEVICE CLEAR"

2191    ********************** TEST 5 ***********************
        TEST TO VERIFY THAT BIT "SILOEN" CAN
        BE SET. THEN VERIFY THAT BIT "SILOEN" CAN
        BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
        VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
        CLEARED BY A "DEVICE CLEAR"

2223    ********************** TEST 6 ***********************
        TEST TO VERIFY THAT BIT "RIE" CAN
        BE SET. THEN VERIFY THAT BIT "RIE" CAN
        BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
        VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
        CLEARED BY A "DEVICE CLEAR"

2255    ********************** TEST 7 ***********************
        TEST TO VERIFY THAT BIT "TIE" CAN
        BE SET. THEN VERIFY THAT BIT "TIE" CAN
        BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
        VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
        CLEARED BY A "DEVICE CLEAR"

2287    ********************** TEST 10 **********************
        THIS TESTS THAT ALL OF THE FOLLOWING
        BITS CAN BE: SET, CLEARED, CLEARED BY "DEVICE CLEAR "
        BITS TESTED ARE:
          TCR0, TCR1, TCR2, TCR3, TCR4, TCR5, TCR6, TCR7

2329    ********************** TEST 11 **********************
        THIS TESTS THAT ALL OF THE FOLLOWING
        BITS CAN BE: SET, CLEARED, CLEARED BY "RESET INSTR *NOT* DEVICE CLEAR "
        BITS TESTED ARE:

2333     DTR0, DTR1, DTR2, DTR3, DTR4, DTR5, DTR6, DTR7
        THIS TEST IS NOT DONE IF MODULE IS 20MA VERSION

2382    ********************** TEST 12 **********************
        THIS TEST PERFORMS RESET TESTING &
        TESTING OF WRITE ONLY OR READ ONLY BIT
         TEST BITS "RDONF, BIT11, BIT10, BIT9, BIT8, BIT2, BIT1
                   BIT0, SILOAL" ARE READ ONLY AND THAT TRDY IS
                   ZERO UNTIL A LINE IS SELECTED AND MSENAB IS SET.
```

```
2419    ********************** TEST 13 ********************************
        THIS TEST PERFORMS RESET TESTING AND
        TESTING OF READ ONLY AND WRITE ONLY BITS
         IN REGISTER DZCSR
        VERIFY THAT "TIE", "SILOEN", "RIE", "MSENAB", "MAINT"
        ARE THE ONLY R/W BITS IN THE DZCSR.
        THEN VERIFY THAT A RESET WILL CLEAR THESE BITS
        THIS TEST ALSO CHECKS BYTE OPERATIONS ON THE CSR

2463    ********************** TEST 14 ********************************
        THIS TEST PERFORMS RESET TESTING AND
        TESTING OF READ ONLY REGISTER DZRBUF
        AND TESTING OF WRITE ONLY REGISTER DZLPR

2489    ********************** TEST 15 ********************************
        THIS TEST PERFORMS RESET TESTING AND
        TESTING OF READ ONLY REGISTER DZMSR
        AND TESTING OF WRITE ONLY REGISTER DZTDR

2516    ********************** TEST 16 ********************************
        VERIFY THAT IF WE ARE IN "STAGGERED" MODE
        THAT SETTING "DTR" FOR A LINE WILL
        BRING UP "RING" AND "CARRIER" FOR THE
        ASSOCIATED LINE IN WHICH WE ARE STAGGERED!
         LINE0 DTR= LINE1 RING AND CARRIER
         LINE1 DTR= LINE0 RING AND CARRIER
         LINE2 DTR= LINE3 RING AND CARRIER
         LINE3 DTR= LINE 4 RING AND CARRIER
                    ETC...

2575    ********************** TEST 17 ********************************
        TEST TO VERIFY THAT IF IN "EXTERNAL"
        MODE; SETTING DTR FOR SELECTED LINES
        WILL BRING UP "CARRIER" AND "RING"
        FOR THAT SAME LINE. NOTE: IF YOU HAVE
        SELECTED MODE AS "EXTERNAL", THE H325 TEST CONNECTER
        MUST BE USED ON ALL SPECIFIED LINES.
        LINES MAY BE SPECIFIED BY SWR03=1
        AND SWR00=1 AT START TIME OR ALTERING
        STATUS MAP.

2622    ********************** TEST 20 ********************************
         THIS TEST VERIFIES THAT TRDY IS SET WHEN A LINE
         IS READY TO BE LOADED, AND THAT THE LINE SPECI-
         FIED IN BITS 8-10 OF DZCSR CORRESPOND
         TO THE LINE SELECTED IN DZTCR

2658    ********************** TEST 21 ********************************
        TEST TO TRANSMIT ONE CHAR AND
        RECEIVE ONE CHAR ON ONE LINE
        AT A TIME. THE CHAR IS "252" AND
        ALL SELECTED LINES WILL BE TURNED ON
        ONE AT A TIME. THIS IS THE FIRST TIME ANY
        DATA IS CHECKED IN THE RECEIVER.
```

```
                USING SWITCH NINE WITH THIS TEST CREATES A TIGHT SCOPE LOOP
                WHICH TRANSMITS A STEADY STREAM OF CHARACTERS.

2749            ********************** TEST 22 ******************************
                 THIS TEST PROVES THAT THE TRANSMITTER TRANSMITS
                CHARACTERS (FLAG MODE)AND THE RECEIVER RECEIVES (FLAG MODE)
                (ONE LINE AT A TIME   BASED UPON VALID LINES)
                THIS IS THE FIRST TIME THAT ALL DATA IS CHECKED

2830            ********************** TEST 23 ******************************
                THIS TEST WILL PROVE THAT EACH RECEIVING LINE CAN
                BE DISABLED BY SETTING THE RCVON BIT TO ZERO
                FOR EACH LINE IN THE LPR REGISTER.  IT ALSO
                VERIFIES THAT MASTER CLEAR WILL ZERO DVALID FOR
                CHARACTERS STORED IN THE SILO.

2915            ********************** TEST 24 ******************************
                THIS TEST WILL PROVE THAT:
                 1) THE TRANSMITTER "BREAK BIT" WORKS
                 2) THE RECEIVER CAN FLAG "FRAMING ERRORS"
                 3) THE RECEIVER CAN FLAG "PARITY ERRORS"
                ONLY ONE LINE AT A TIME WILL BE EXERCISED.
                THIS TEST WILL NOT BE  EXERCISED UNLESS
                CONNECTED BY AN H325, H3271, OR H3190 CONNECTOR

2982            ********************** TEST 25 ******************************
                 THIS TEST VERIFIES THAT THE DEVICE DOES NOT INTERRUPT
                WHILE THE PROCESSOR STATUS IS SET EXACTLY
                TO WHAT THE DZ11 PRIORITY IS SET TO.
                DEFAULT PRIORITY IS AT  5 (240).

3051            ********************** TEST 26 ******************************
                 THIS TEST VERIFIES THAT THE DEVICE DOES INTERRUPT
                WHILE THE PROCESSOR STATUS IS SET TO EXACTLY
                ONE LEVEL LOWER THAN THE DZ11. DZ11 PRIORITY
                DEFAULT TO LEVEL 5 MINUS ONE LEVEL IS LEVEL 4.

3124            ********************** TEST 27 ******************************
                THIS TEST VERIFIES THAT THE RECEIVER WILL
                INTERRUPT BEFORE THE TRANSMITTER EVEN
                THOUGH THE TRANSMITTER WAS ENABLED
                FIRST.  SET PS TO LEVEL 7;
                GET RDONE AND TRDY TO SET;
                SET TX IE AND RX IE;
                CLEAR PS AND EXPECT RX TO INTERRUPT FIRST

3234            ********************** TEST 30 ******************************
                TEST TO VERIFY THAT 'RDONE DOES NOT SET
                IF THE SCANNER IS DISABLED.
                TURN ON SCANNER, WAIT FOR TRDY,
                TURN OFF SCANNER, TRANSMIT A CHARACTER
                'RDONE SHOULD NOT SET.
```

```
3280    ********************** TEST 31 ********************************
        THIS TEST VERIFIES OVERRUN AND SILO ALARM
        ONE LINE AT A TIME  - BASED UPON VALID LINES
        AS EACH OF THE FIRST 16 CHARS ARE SENT; SILO ALARM IS
        TESTED TO BE CLEARED. ON THE 16TH CHAR THE PROGRAM THEN

3285    EXPECTS SILO ALARM TO SET. THEN, THE ENTIRE
        SILO IS FILLED AND AN OVERRUN IS EXPECTED ON THE 65TH
        CHAR PULLED OUT OUT THE SILO.
        USING SWITCH NINE FOR THIS TEST SENDS 20. CHARACTERS
        ON DZ LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
        USED TO SCOPE SILO ALARM PULSES, ETC.

3415    ********************** TEST 32 ********************************
        THIS TEST THAT "SILO ENABLE" WILL INHIBIT
        RECEIVER INTERRUPTS AND THAT ON THE
        16TH CHAR THAT "SILO ALARM" WILL CAUSE AN
        INTERRUPT WITH "RIE" SET.
        THIS WILL DO ALL SELECTED LINES ONE AT A TIME.

3500    ********************** TEST 33 ********************************
        THIS TEST RUNS ALL LINES FULL BORE
        BASED UPON QUALIFIED LINES
        ..THIS IS AN INTERRUPT TEST ON THE RECEIVER AND
        TRANSMITTER

3644    ********************** TEST 34 ********************************
        DZ11 RELATIVE TIMING TEST.
        EACH SELECTED LINE WILL IN TURN RUN 16. CHARS
        AT ALL BAUD RATES AND THEN THE HIGHEST BAUD
        WITH ALL CHAR LENGTHS. EACH NEW PARAMETER SHOULD
        DECREASE IN TIME FROM THE PREVIOUS PARAMETERS SELECTED.
        THE TIME IS CHECKED AGAINST THE LAST PARAMETER USED
         AND A LOWER TIME IS EXPECTED ON THE CURRENT PARAMETER.
        PARAMETERS ARE:
          EIGHT BITS/PER/CHAR - TWO STOP BITS AT
                50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000
                2400, 3600, 4800, 7200, 9600 BAUD.
          19.2 K BAUD - TWO STOP BITS AT
                SEVEN, SIX, FIVE BITS/PER/CHAR.
        AFTER EACH LINE HAS FINISHED ALL THE ABOVE PARAMETERS
        THE NEXT SELECTED LINE IS THE TESTED.

3743    ********************** TEST 35 ********************************
         THIS TEST VERIFIES THAT EVEN PARITY WORKS
         FOR ALL ODD LINES SELECTED AND THAT ODD PARITY WORKS FOR ALL
         EVEN LINES SELECTED.
        THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
        THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
        THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
        YOU ARE IN "STAGGERED" MODE.
        40(8) CHARS ARE USED FOR THIS TEST.
        ALL SELECTED LINES WILL BE ENABLED
        AT THE SAME TIME!
```

```
3800    ********************** TEST 36 ********************************
        THIS TEST VERIFIES THAT ODD PARITY WORKS FOR ALL ODD LINES
         SELECTED AND THAT EVEN PARITY WORKS FOR ALL EVEN LINES SELECTED
        THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
        THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
        THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
        YOU ARE IN "STAGGERED" MODE.
        40(8) CHARS ARE USED FOR THIS TEST.
        ALL SELECTED LINES WILL BE ENABLED
        AT THE SAME TIME!

3985    STARTING PROCEDURE
        LOAD PROGRAM
        LOAD ADDRESS 000210
        PRESS START
        PROGRAM WILL TYPE DZ11 ECHO/CABLE TEST
        PROGRAM WILL TYPE  WHICH TEST- ECHO OR CABLE
        TYPE IN  E  OR  C   RESPECTIVELY
        PROGRAM WILL TYPE "VECTOR ADDRESS-"
        TYPE IN THE ADDRESS OF THE RECEIVER INTERRUPT VECTOR
        FOR THE DZ11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
        PROGRAM WILL TYPE "CONTROL REGISTER ADDRESS-"
        TYPE IN THE ADDRESS OF THE SYSTEM CONTROL REGISTER
        FOR THE DZ11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
        PROGRAM WILL TYPE "LINE NUMBER-"
        TYPE IN THE LINE NUMBER TO BE TESTED (IN OCTAL)
         FOLLOWED BY <CARRIAGE RETURN>
        PROGRAM WILL TYPE "BAUD RATE-"
        TYPE IN THE BAUD RATE OF THE DZ11 TERMINAL
         FOLLOWED BY <CARRIAGE RETURN>
        THE FOLLOWING BAUD RATES ARE ACCEPTED IN DECIMAL
                        50
                        75
                        110
                        135     (ROUNDED OFF   134.5)
                        150
                        300
                        600
                        1200
                        1800
                        2000
                        2400
                        3600
                        4800
                        7200
                        9600
        ALL OTHERS ARE REJECTED

4022    PROGRAM WILL TYPE "ECHO"  OR  "CABLE TEST" TO INDICATE THAT TESTING HAS STARTE
```

```
4208    TEST TO VERIFY THAT SETTING DTR FOR A GIVEN LINE
        WILL BRING UP "CO" AND "RING" FOR THE SAME LINE
        THE DIST PNL MUST HAVE JUMPER FROM DTR TO RQST TO SEND
        IN ORDER FOR THIS TEST TO WORK!

ENDCOM    132#
ERROR      26#  2094  2112  2125  2144  2150  2157  2176  2182  2189  2208  2214  222

ESCAPE    132#
GETPRI    132#
GETSWR    132#
MULT      132#
NEWTST    132#  2059  2102  2134  2166  2198  2230  2262  2293  2336  2390  2428  246

PASEND      1#  1095
POP       132#  1349  1350  1736  1737
PRGEND      1#  1082
PRGFRT      1#     3
PUSH      132#  1310  1312  1333  1717  1723
REPORT      1#   132#
SC          1#  1161
SCOPE      27#  1095  2060  2103  2135  2167  2199  2231  2263  2294  2337  2391  242

SC1         1#  1201
SETPRI    132#
SETUP     132#
SKIP      132#
SLASH     132#
SPACE     132#
STARS     132#   332   354   405   408   543   545   552  1089  1150  1226  1305  171

SWRSU     132#
TYPBIN    132#
TYPDEC    132#
TYPNAM    132#
TYPNUM    132#
TYPOCS    132#
TYPOCT    132#
TYPTXT    132#
$BUFFE      1#  1787
$CYCLE      1#  1814
$EOP        1#  1082
$GETFL      1#   896
$GETPA      1#   835   848   861   903   959   992  1050  1862
```

$HEADE	1#	13											
$INTSE	1#	3004	3027	3073	3098	3785	3841						
$JUNK	1#	568	575	582	589	596	603	610	617	624	631	638	64
$LINEU	1#	2673	2760	2992	3061	3137	3245	3297	3427				
$LVLTS	1#	2982	3051										
$MRESE	1#	2673	2760	2931	2992	3049	3061	3121	3137	3245	3297	3427	351
$MRR	1#	2382											
$MRRW	1#	2127	2159	2191	2223	2255							
$MRWD	1#	2463	2489										
$MSG	1#	1753											
$PARTS	1#	3743	3800										
$SCOPE	1#	1144											
$SETFL	1#	1759											
$STAG	1#	2700	2793	2951	3204	3358							
$STAGF	1#												
$TCR	1#	2287	2329										
$TLINE	1#	2622											
$TRPDE	1#	689	691	693	695	697	699	701	703	705	707	709	71
$TSTN	1#	2058	2101	2133	2165	2197	2229	2261	2292	2335	2389	2427	246
$UNIBU	1#	2054											
$VARIA	1#	349											
$XZ	1#	2054	2058	2097	2101	2127	2133	2159	2165	2191	2197	2223	222
$$CMRE	352#	391	392	393	394	395	396						
$$CMTM	352#	397	398	399	400								
$$ESCA	132#												
$$NEWT	132#	2059	2102	2134	2166	2198	2230	2262	2293	2336	2390	2428	246
$$SKIP	132#												
.EQUAT	1#	22											
.HEADE	1#	3											
.SETUP	1#												
.$ACT1	1#	330											
.$APTB	1#	406#											
.$APTH	1#	541											
.$APTY	1#	1303											
.$CATC	1#												
.$CMTA	1#	352											
.$EOP	1#	1087											
.$ERRO	1#												
.$POWE	1#	1711											
.$SCOP	1#	1148											
.$TRAP	1#												
.$TYPE	1#	1223											

```
. ABS.  031110     000

ERRORS DETECTED:  0

DZDZAE,DZDZAE/SOL/CRF/NL:TOC=DZDZAE.P11
RUN-TIME: 29 20 2 SECONDS
RUN-TIME RATIO: 246/52=4.6
CORE USED:  36K  (71 PAGES)
```

```
                        .TITLE  MD-11-DZDZA-E
                        ;*COPYRIGHT (C) 1977
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*
000001                  $TN=1
                                ;STARTING PROCEDURE
                                ;LOAD PROGRAM
                                ;LOAD ADDRESS 000200
                                ;PRESS START
                                ;PROGRAM WILL TYPE "MAINDEC-11-DZDZAE/<200>/EIGHT LINE ASYNC MUX TESTS"
                                ;PROGRAM WILL TYPE "RUNNING" TO INDICATE THAT TESTING HAS STARTED
                                ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
                                ;AND THEN RESUME TESTING

                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1120 ***
001120                  STACK=  1120
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE FOR LINE FEED
000015                  CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
000007                  PC=     %7              ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
```

```
                        000140          PR3=    140             ;;PRIORITY LEVEL 3
                        000200          PR4=    200             ;;PRIORITY LEVEL 4
                        000240          PR5=    240             ;;PRIORITY LEVEL 5
                        000300          PR6=    300             ;;PRIORITY LEVEL 6
                        000340          PR7=    340             ;;PRIORITY LEVEL 7

                                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
                        100000          SW15=   100000
                        040000          SW14=   40000
                        020000          SW13=   20000
                        010000          SW12=   10000
                        004000          SW11=   4000
                        002000          SW10=   2000
                        001000          SW09=   1000
                        000400          SW08=   400
                        000200          SW07=   200
                        000100          SW06=   100
                        000040          SW05=   40
                        000020          SW04=   20
                        000010          SW03=   10
                        000004          SW02=   4
                        000002          SW01=   2
                        000001          SW00=   1
                                        .EQUIV  SW09,SW9
                                        .EQUIV  SW08,SW8
                                        .EQUIV  SW07,SW7
                                        .EQUIV  SW06,SW6
                                        .EQUIV  SW05,SW5
                                        .EQUIV  SW04,SW4
                                        .EQUIV  SW03,SW3
                                        .EQUIV  SW02,SW2
                                        .EQUIV  SW01,SW1
                                        .EQUIV  SW00,SW0

                                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                        100000          BIT15=  100000
                        040000          BIT14=  40000
                        020000          BIT13=  20000
                        010000          BIT12=  10000
                        004000          BIT11=  4000
                        002000          BIT10=  2000
                        001000          BIT09=  1000
                        000400          BIT08=  400
                        000200          BIT07=  200
                        000100          BIT06=  100
                        000040          BIT05=  40
                        000020          BIT04=  20
                        000010          BIT03=  10
                        000004          BIT02=  4
                        000002          BIT01=  2
                        000001          BIT00=  1
                                        .EQUIV  BIT09,BIT9
                                        .EQUIV  BIT08,BIT8
                                        .EQUIV  BIT07,BIT7
                                        .EQUIV  BIT06,BIT6
                                        .EQUIV  BIT05,BIT5
```

```
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;"TRAP" TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR


                        ;INSTRUCTION DEFINITIONS
                        ;----------------------

005746                  PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
005726                  POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
010046                  PUSHR0=10046    ;SAVE R0 ON STACK
012600                  POPR0=12600     ;RESTORE R0 FROM STACK
024646                  PUSH2SP=24646   ;DECREMENT STACK TWICE
022626                  POP2SP=22626    ;INCREMENT STACK TWICE

                        ;DZ11 CONTROL AND STATUS REGISTER DEFINITIONS
                        ;(DZCSR)        BIT DEFINITIONS
                        ;-------------------------------

000010                  MAINT = BIT3    ;MAINTENANCE MODE ENABLE
000020                  DCLR=BIT4       ;DEVICE CLEAR
000040                  MSENAB=BIT5     ;MASTER SCAN ENABLE
000100                  RIE=BIT6        ;RECEIVER INTERRUPT ENABLE
000200                  RDONE=BIT7      ;RECEIVER DONE
010000                  SILOEN= BIT12   ;SILO ALARM ENABLE
020000                  SILOAL = BIT13  ;SILO ALARM
040000                  TIE=BIT14       ;TRANSMITTER INTERRUPT ENABLE
100000                  TRDY=BIT15      ;TRANSMITTER READY

                        ;DZCSR WORD DEFINITIONS
                        ;----------------------
000000                  TL0=0           ;TRANSMIT LINE 0
000400                  TL1=BIT8        ;TRANSMIT LINE 1
001000                  TL2=BIT9        ;TRANSMIT LINE 2
001400                  TL3=BIT9!BIT8   ;TRANSMIT LINE 3
002000                  TL4=BIT10       ;TRANSMIT LINE 4
002400                  TL5=BIT10!BIT8  ;TRANSMIT LINE 5
003000                  TL6=BIT10!BIT9  ;TRANSMIT LINE 6
003400                  TL7=BIT10!BIT9!BIT8 ;TRANSMIT LINE 7
```

```
                                ;DZRBUF BIT DEFINITIONS
                                ;----------------------

010000                  PARER=BIT12     ;PARITY ERROR
020000                  FRMERR=BIT13    ;FRAME ERROR
040000                  OVRRUN=BIT14    ;OVERRUN ERROR
100000                  DVALID=BIT15    ;DATA VALID

                                ;DZRBUF WORD DEFINITIONS
                                ;-----------------------

000000                  RL0=0           ;RECEIVER LINE 0
000400                  RL1=BIT8        ;RECEIVER LINE 1
001000                  RL2=BIT9        ;RECEIVER LINE 2
001400                  RL3=BIT9!BIT8   ;RECEIVER LINE 3
002000                  RL4=BIT10       ;RECEIVER LINE 4
002400                  RL5=BIT10!BIT8  ;RECEIVER LINE 5
003000                  RL6=BIT10!BIT9  ;RECEIVER LINE 6
003400                  RL7=BIT10!BIT9!BIT8 ;RECEIVER LINE 7

                                ;DZLPR WORD DEFINITIONS
                                ;----------------------

000000                  LP0=0           ;LINE PARAMETER 0
000001                  LP1=BIT0        ;LINE PARAMETER 1
000002                  LP2=BIT1        ;LINE PARAMETER 2
000003                  LP3=BIT1!BIT0   ;LINE PARAMETER 3
000004                  LP4=BIT2        ;LINE PARAMETER 4
000005                  LP5=BIT2!BIT0   ;LINE PARAMETER 5
000006                  LP6=BIT2!BIT1   ;LINE PARAMETER 6
000007                  LP7=BIT2!BIT1!BIT0      ;LINE PARAMETER 7

000000                  FIVE=0          ;FIVE BITS/CHAR,1 STOP BIT
000010                  SIX=BIT3        ;SIX BITS/CHAR,1 STOP BIT
000020                  SEVEN=BIT4      ;SEVEN BITS/CHAR,1 STOP BIT
000030                  EIGHT=BIT4!BIT3 ;EIGHT BITS/CHAR,1 STOP BIT
000040                  FIVES=BIT5      ;FIVE BITS/CHAR,2 STOP BITS
000050                  SIXS=BIT5!BIT3  ;SIX BITS/CHAR,2 STOP BITS
000060                  SEVENS=BIT5!BIT4        ;SEVEN BITS/CHAR, 2 STOP BITS
000070                  EIGHTS=BIT5!BIT4!BIT3   ;EIGHT BITS/CHAR, 2 STOP BITS

000100                  PARITY=BIT6             ;PARITY ENABLED
000200                  ODDPAR=BIT7             ;ODD PARITY ENABLED
000000                  ONESTOP=0               ;ONE STOP BIT ENABLED
000040                  TWOSTOP=BIT5            ;TWO STOP BITS ENABLED
000000                  EVEPAR=0                ;EVEN PARITY ENABLED
010000                  RCVON=BIT12             ;ENABLE RECEIVER (RECEIVER ON)

000000                  S50=0                   ;SPEED 50 BAUD
000400                  S75=BIT8                ;SPEED 75 BAUD
001000                  S110=BIT9               ;SPEED 110 BAUD
001400                  S134=BIT9!BIT8          ;SPEED 134.5 BAUD
002000                  S150=BIT10              ;SPEED 150 BAUD
002400                  S300=BIT10!BIT8         ;SPEED 300 BAUD
003000                  S600=BIT10!BIT9         ;SPEED 600 BAUD
```

```
                        S1200=BIT10!BIT9!BIT8   ;SPEED 1200 BAUD
                        S1800=BIT11             ;SPEED 1800 BAUD
                        S2000=BIT11!BIT8        ;SPEED 2000 BAUD
                        S2400=BIT11!BIT9        ;SPEED 2400 BAUD
                        S3600=BIT11!BIT9!BIT8   ;SPEED 3600 BAUD
                        S4800=BIT11!BIT10       ;SPEED 4800 BAUD
                        S7200=BIT11!BIT10!BIT8  ;SPEED 7200 BAUD
                        S9600=BIT11!BIT10!BIT9  ;SPEED 9600 BAUD
                        S19200=BIT11!BIT10!BIT9!BIT8    ;SPEED 19200 BAUD

                                ;DZTCR BIT DEFINITIONS
                                ;---------------------
000001                  TCR0=BIT0               ;TCR0
000002                  TCR1=BIT1               ;TCR1
000004                  TCR2=BIT2               ;TCR2
000010                  TCR3=BIT3               ;TCR3
000020                  TCR4=BIT4               ;TCR4
000040                  TCR5=BIT5               ;TCR5
000100                  TCR6=BIT6               ;TCR6
000200                  TCR7=BIT7               ;TCR7
000400                  DTR0=BIT8               ;DTR0
001000                  DTR1=BIT9               ;DTR1
002000                  DTR2=BIT10              ;DTR2
004000                  DTR3=BIT11              ;DTR3
010000                  DTR4=BIT12              ;DTR4
020000                  DTR5=BIT13              ;DTR5
040000                  DTR6=BIT14              ;DTR6
100000                  DTR7=BIT15              ;DTR7

                                ;DZMSR BIT DEFINITIONS
                                ;---------------------
000001                  RING0=BIT0              ;RING INDICATED ON LINE 0
000002                  RING1=BIT1              ;RING INDICATED ON LINE 1
000004                  RING2=BIT2              ;RING INDICATED ON LINE 2
000010                  RING3=BIT3              ;RING INDICATED ON LINE 3
000020                  RING4=BIT4              ;RING INDICATED ON LINE 4
000040                  RING5=BIT5              ;RING INDICATED ON LINE 5
000100                  RING6=BIT6              ;RING INDICATED ON LINE 6
000200                  RING7=BIT7              ;RING INDICATED ON LINE 7
000400                  CO0=BIT8                ;CARRIER PRESENT ON LINE 0
001000                  CO1=BIT9                ;CARRIER PRESENT ON LINE 1
002000                  CO2=BIT10               ;CARRIER PRESENT ON LINE 2
004000                  CO3=BIT11               ;CARRIER PRESENT ON LINE 3
010000                  CO4=BIT12               ;CARRIER PRESENT ON LINE 4
020000                  CO5=BIT13               ;CARRIER PRESENT ON LINE 5
040000                  CO6=BIT14               ;CARRIER PRESENT ON LINE 6
100000                  CO7=BIT15               ;CARRIER PRESENT ON LINE 7

                                ;DZTDR BIT DEFINITIONS
                                ;---------------------

000400                  BRK0=BIT8               ;BREAK FOR LINE 0
001000                  BRK1=BIT9               ;BREAK FOR LINE 1
002000                  BRK2=BIT10              ;BREAK FOR LINE 2
004000                  BRK3=BIT11              ;BREAK FOR LINE 3
010000                  BRK4=BIT12              ;BREAK FOR LINE 4
```

```
281   020000   BRK5=BIT13          ;BREAK FOR LINE 5
282   040000   BRK6=BIT14          ;BREAK FOR LINE 6
283   100000   BRK7=BIT15          ;BREAK FOR LINE 7
284
285
286            ;TABLE OF LOOP AROUND FUNCTIONS (H325)
287            ;
288            ;       -----------------
289            ;       I               ↑
290            ;       V               ↑
291            ;       REC             TRANS
292            ;       DATA            DATA
293            ;
294            ;       -----------------
295            ;       I               ↑
296            ;       V               ↑
297            ;       CO              RTS
298            ;
299            ;
300            ;       -----------------
301            ;       I               ↑
302            ;       V               ↑
303            ;       RING            DTR
```

```
                                ;:*****************************************************************************
                                ;-----------------------------------------------------------------------------
                                ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
                                ;THE STANDARD "TRAP CATCHER" IS PLACED
                                ;BETWEEN ADDRESS 0 TO ADDRESS 776.
                                ;IT LOOKS LIKE "PC+2 HALT".
                                ;-----------------------------------------------------------------------------
                                ;:*****************************************************************************

                        000000  .=0
                                ;STANDARD INTERRUPT VECTORS
                                ;-------------------------

                        000010  .=10
000010  010766                  SET.PS                  ;FAKE "MTPS" INSTRUCTION TRAP
000012  000340                  PR7                     ;MAKE SURE PS IS PRIORITY 7

                        000020  .=20
000020  004772                  .SCOPE                  ;SCOPE LOOP HANDLER
000022  000340                  PR7                     ;HANDLE AT PRIORITY 7
000024  007646                  $PWRDN                  ;POWER FAIL HANDLER
000026  000340                  340                     ;SERVICE AT PRIORITY LEVEL 7
000030  006736                  $ERROR                  ;ERROR HANDLER
000032  000340                  340                     ;SERVICE AT PRIORITY LEVEL 7
000034  006630                  .TRPSRV                 ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                  340                     ;SERVICE AT PRIORITY LEVEL 7
                        .SBTTL  ACT11 HOOKS

                                ;:*****************************************************************
                                ;HOOKS REQUIRED BY ACT11
                        000040          $SVPC=.                 ;SAVE PC
                        000046          .=46
000046  004726                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                        000052          .=52
000052  000000                  .WORD   0               ;;2)SET LOC.52 TO ZERO
                        000040          .=$SVPC                 ;; RESTOPE PC

                        000174  .=174
000174  000000          DISPREG:0                       ;SOFTWARE DISPLAY REGISTER FOR SWITCHLESS 11S
000176  000000          SWREG:  0                       ;SOFTARE SWITCH REGISTER FOR SWITCHLESS 11S
                        000200  .=200
000200  000137  002150          JMP     .START          ;GO TO START OF PROGRAM
                        000210  .=210
000210  000137  023770          JMP     XSTART          ;GOTO CABLE TEST/ECHO TEST

                        001000  .=1000
001000  005200  040515  047111  MTITLE: .ASCIZ  <200><12>/MAINDEC-11-DZDZAE/<200>/EIGHT LINE ASYNC MUX TESTS/<200>
(2)
```

```
                                .SBTTL  COMMON TAGS

                                ;;**********************************************************************
                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.

                        001120          .=1120
001120                  $CMTAG:                         ;;START OF COMMON TAGS
001120  000000                  .WORD   0
001122  000             $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001123  000             $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001124  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001126  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001130  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001132  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001134  000             $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001135  001             $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001136  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001140  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001142  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001144  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001146  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001150  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001152  000000                  .WORD   0
001154  000             $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001155  000             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001156  000000                  .WORD   0
001160  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001162  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001164  177560          $TKS:   177560                  ;;TTY KBD STATUS
001166  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001170  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001172  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001174  000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001175  002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001176  012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001177  000             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001200  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001202  000000          $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001204  000000          $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001206  000000          $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001210  000000          $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001212  000000          $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001214  000000          $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001216  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001220  000000          $TMP1:  .WORD   0               ;;USER DEFINED
001222  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001224  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001226  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001230  077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001231  015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001232  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;**********************************************************************
                                .SBTTL  APT MAILBOX-ETABLE
```

```
                                ;;*****************************************************************
                                .EVEN
001234                          $MAIL:                          ;;APT MAILBOX
001234  000000                  $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001236  000000                  $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001240  000000                  $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001242  000000                  $PASS:  .WORD   APASS           ;;PASS COUNT
001244  000000                  $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001246  000000                  $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
001250  000000                  $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
001252  000000                  $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
001254                          $ETABLE:                        ;;APT ENVIRONMENT TABLE
001254  000                     $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
001255  000                     $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
001256  000000                  $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
001260  000000                  $USWR:  .WORD   AUSWR           ;;USER SWITCHES
001262  000000                  $CPUOP: .WORD   ACPUOP          ;;CPU TYPE,OPTIONS
                                ;*                              BITS 15-11=CPU TYPE
                                ;*                                      11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                ;*                                      11/70=06,PDQ=07,Q=10
                                ;*                              BIT 10=REAL TIME CLOCK
                                ;*                              BIT  9=FLOATING POINT PROCESSOR
                                ;*                              BIT  8=MEMORY MANAGEMENT
001264  000                     $MAMS1: .BYTE   AMAMS1          ;;HIGH ADDRESS,M.S. BYTE
001265  000                     $MTYP1: .BYTE   AMTYP1          ;;MEM. TYPE,BLK#1
                                ;*                              MEM.TYPE BYTE   --  (HIGH BYTE)
                                ;*                                      900 NSEC CORE=001
                                ;*                                      300 NSEC BIPOLAR=002
                                ;*                                      500 NSEC MOS=003
001266  000000                  $MADR1: .WORD   AMADR1          ;;HIGH ADDRESS,BLK#1
                                ;*                                MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001270  000                     $MAMS2: .BYTE   AMAMS2          ;;HIGH ADDRESS,M.S. BYTE
001271  000                     $MTYP2: .BYTE   AMTYP2          ;;MEM.TYPE,BLK#2
001272  000000                  $MADR2: .WORD   AMADR2          ;;MEM.LAST ADDRESS,BLK#2
001274  000                     $MAMS3: .BYTE   AMAMS3          ;;HIGH ADDRESS,M.S.BYTE
001275  000                     $MTYP3: .BYTE   AMTYP3          ;;MEM.TYPE,BLK#3
001276  000000                  $MADR3: .WORD   AMADR3          ;;MEM.LAST ADDRESS,BLK#3
001300  000                     $MAMS4: .BYTE   AMAMS4          ;;HIGH ADDRESS,M.S.BYTE
001301  000                     $MTYP4: .BYTE   AMTYP4          ;;MEM.TYPE,BLK#4
001302  000000                  $MADR4: .WORD   AMADR4          ;;MEM.LAST ADDRESS,BLK#4
001304  000000                  $VECT1: .WORD   AVECT1          ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001306  000000                  $VECT2: .WORD   AVECT2          ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001310  160010                  $BASE:  .WORD   ABASE           ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001312  000000                  $DEVM:  .WORD   ADEVM           ;;DEVICE MAP
001314  000000                  $CDW1:  .WORD   ACDW1           ;;CONTROLLER DESCRIPTION WORD#1
001316  000000                  $CDW2:  .WORD   ACDW2           ;;CONTROLLER DESCRIPTION WORD#2
001320  000000                  $DDW0:  .WORD   ADDW0           ;;DEVICE DESCRIPTOR WORD#0
001322  000000                  $DDW1:  .WORD   ADDW1           ;;DEVICE DESCRIPTOR WORD#1
001324  000000                  $DDW2:  .WORD   ADDW2           ;;DEVICE DESCRIPTOR WORD#2
001326  000000                  $DDW3:  .WORD   ADDW3           ;;DEVICE DESCRIPTOR WORD#3
001330  000000                  $DDW4:  .WORD   ADDW4           ;;DEVICE DESCRIPTOR WORD#4
001332  000000                  $DDW5:  .WORD   ADDW5           ;;DEVICE DESCRIPTOR WORD#5
001334  000000                  $DDW6:  .WORD   ADDW6           ;;DEVICE DESCRIPTOR WORD#6
001336  000000                  $DDW7:  .WORD   ADDW7           ;;DEVICE DESCRIPTOR WORD#7
001340  000000                  $DDW8:  .WORD   ADDW8           ;;DEVICE DESCRIPTOR WORD#8
001342  000000                  $DDW9:  .WORD   ADDW9           ;;DEVICE DESCRIPTOR WORD#9
```

```
   464  001344  000000                  $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
   465  001346  000000                  $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
   466  001350  000000                  $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
   467  001352  000000                  $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
   468  001354  000000                  $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
   469  001356  000000                  $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
   470
   471
   472  001360                          $ETEND:
   473
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT


001360                          $ERRTB:

                                        ;PROGRAM CONTROL PARAMETERS
                                        ;--------------------------

001360  000000                  NEXT:   0                       ;ADDRESS OF NEXT TEST TO BE EXECUTED
001362  000000                  LOCK:   0                       ;ADDRESS FOR LOCK ON CURRENT DATA

                                        ;PROGRAM VARIABLES
                                        ;-----------------

001364  000377                  LINE:   377                     ;DEFAULT ALL EIGHT LINES RUNNING
001366  017470                  PAR:    17470                   ;PARAMETERS: 8 BITS/CHAR,2 STOP BITS,19200 BAUD,NO PARIT
001370  000000                  MODE:   0                       ;DEFAULT MAINTENANCE MODE
001372  000000                  SAVLIN: 0                       ;LINE NUMBER
001374  000000                  XMTLIN: 0                       ;TRANSMISSION LINE NUMBER
001376  000000                  XMTCNT: 0                       ;COUNT OF WORDS IN A TRANSMISSION PATTERN
001400  000000                  REGIST: 0                       ;DEVICE ADDRESS STORAGE LOCATION
001402  000000                  SAVPC:  0                       ;PROGRAM COUNTER STORAGE
001404  000001                  DZACTV: .BLKW   1               ;*DZ11'S SELECTED ACTIVE.
001406  000001                  RUN:    1                       ;*POINTER ONE PAST RUNNING DEVICE.
001410  000001                  DZNUM:  .BLKB 1                 ;*OCTAL NUMBER OF DZ11'S.
001411     001                  SAVNUM: .BYTE 1                 ;*WORKABLE NUMBER.
                                .EVEN
001412  001500                  ACTIVE: DZ.MAP                  ;TABLE POINTER.
```

```
                                        ;PROGRAM CONTROL FLAGS
                                        ;---------------------

001414  000                   EIAFLG: .BYTE   0               ;0=EIA  100000=20MA
001415  000                   INIFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG
001416  000                   HDRFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG FOR HEADER MAP
001417  000                   MNTFLG: .BYTE   0               ;MAINTENANCE BIT SET FLAG
001420  000                   DONFLG: .BYTE   0               ;TRANSMISSION COMPLETION FLAG
        001422                .EVEN
                              ;DATA VARIABLES
001422  000000                TD0:    .WORD   0
001424  000000                TD1:    .WORD   0
001426  000000                TD2:    .WORD   0
001430  000000                TD3:    .WORD   0
001432  000000                TD4:    .WORD   0
001434  000000                TD5:    .WORD   0
001436  000000                TD6:    .WORD   0
001440  000000                TD7:    .WORD   0
001442  000000                TR0:    .WORD   0
001444  000000                TR1:    .WORD   0
001446  000000                TR2:    .WORD   0
001450  000000                TR3:    .WORD   0
001452  000000                TR4:    .WORD   0
001454  000000                TR5:    .WORD   0
001456  000000                TR6:    .WORD   0
001460  000000                TR7:    .WORD   0
001462                        STOP:
                              .SBTTL  APT PARAMETER BLOCK

                              ;;*****************************************************************
                              ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                              ;;*****************************************************************
        001462                        .$X=.     ;;SAVE CURRENT LOCATION
        000024                        .=24      ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                        200       ;;FOR APT START UP
        000044                        .=44      ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  001462                        $APTHDR   ;;POINT TO APT HEADER BLOCK
        001462                        .=.$X     ;;RESET LOCATION COUNTER
                              ;;*****************************************************************
                              ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                              ;INTERFACE SPEC.

001462                        $APTHD:
001462  000000                $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001464  001234                $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001466  000132                $TSTM:  .WORD   90.     ;;RUN TIM OF LONGEST TEST
001470  000137                $PASTM: .WORD   95.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001472  000137                $UNITM: .WORD   95.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001474  000052                        .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
                              ;DZ11 STATUS TABLE AND ADDRESS ASSIGNMENTS
                              ;-----------------------------------------

        001500                .=1500
001500                        DZ.MAP:
```

```
001500  000001          DZCR0:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 0
001502  000001          DZVC0:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 0
001504  000001          DZLV0:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001506  000001          LINE0:  .BLKW   1       ;ALL LINES SELECTED
001510  000001          PAR0:   .BLKW   1       ;PARAMETERS
001512  000001          MANT0:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

001514  000001          DZCR1:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 1
001516  000001          DZVC1:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 1
001520  000001          DZLV1:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001522  000001          LINE1:  .BLKW   1       ;ALL LINES SELECTED
001524  000001          PAR1:   .BLKW   1       ;PARAMETERS
001526  000001          MANT1:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

001530  000001          DZCR2:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 2
001532  000001          DZVC2:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 2
001534  000001          DZLV2:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001536  000001          LINE2:  .BLKW   1       ;ALL LINES SELECTED
001540  000001          PAR2:   .BLKW   1       ;PARAMETERS
001542  000001          MANT2:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

001544  000001          DZCR3:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 3
001546  000001          DZVC3:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 3
001550  000001          DZLV3:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001552  000001          LINE3:  .BLKW   1       ;ALL LINES SELECTED
001554  000001          PAR3:   .BLKW   1       ;PARAMETERS
001556  000001          MANT3:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

001560  000001          DZCR4:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 4
001562  000001          DZVC4:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 4
001564  000001          DZLV4:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001566  000001          LINE4:  .BLKW   1       ;ALL LINES SELECTED
001570  000001          PAR4:   .BLKW   1       ;PARAMETERS
001572  000001          MANT4:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

001574  000001          DZCR5:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 5
001576  000001          DZVC5:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 5
001600  000001          DZLV5:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001602  000001          LINE5:  .BLKW   1       ;ALL LINES SELECTED
001604  000001          PAR5:   .BLKW   1       ;PARAMETERS
001606  000001          MANT5:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

001610  000001          DZCR6:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 6
001612  000001          DZVC6:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 6
001614  000001          DZLV6:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001616  000001          LINE6:  .BLKW   1       ;ALL LINES SELECTED
001620  000001          PAR6:   .BLKW   1       ;PARAMETERS
001622  000001          MANT6:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

001624  000001          DZCR7:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 7
001626  000001          DZVC7:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 7
001630  000001          DZLV7:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001632  000001          LINE7:  .BLKW   1       ;ALL LINES SELECTED
001634  000001          PAR7:   .BLKW   1       ;PARAMETERS
001636  000001          MANT7:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
```

```
                        ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 10
                        ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 10
                        ;PRIORITY LEVEL AND EIA FLAG SELECTOR
                        ;ALL LINES SELECTED
                        ;PARAMETERS
                        ;MAINTENANCE MODE FOR THIS DEVICE

                        ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 11
                        ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 11
                        ;PRIORITY LEVEL AND EIA FLAG SELECTOR
                        ;ALL LINES SELECTED
                        ;PARAMETERS
                        ;MAINTENANCE MODE FOR THIS DEVICE

                        ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 12
                        ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 12
                        ;PRIORITY LEVEL AND EIA FLAG SELECTOR
                        ;ALL LINES SELECTED
                        ;PARAMETERS
                        ;MAINTENANCE MODE FOR THIS DEVICE

                        ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 13
                        ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 13
                        ;PFIORITY LEVEL AND EIA FLAG SELECTOR
                        ;ALL LINES SELECTED
                        ;PARAMETERS
                        ;MAINTENANCE MODE FOR THIS DEVICE

                        ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 14
                        ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 14
                        ;PRIORITY LEVEL AND EIA FLAG SELECTOR
                        ;ALL LINES SELECTED
                        ;PARAMETERS
                        ;MAINTENANCE MODE FOR THIS DEVICE

                        ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 15
                        ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 15
                        ;PRIORITY LEVEL AND EIA FLAG SELECTOR
                        ;ALL LINES SELECTED
                        ;PARAMETERS
                        ;MAINTENANCE MODE FOR THIS DEVICE

                        ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 16
                        ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 16
                        ;PRIORITY LEVEL AND EIA FLAG SELECTOR
                        ;ALL LINES SELECTED
                        ;PARAMETERS
                        ;MAINTENANCE MODE FOR THIS DEVICE

                        ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 17
                        ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 17
                        ;PRIORITY LEVEL AND EIA FLAG SELECTOR
                        ;ALL LINES SELECTED
                        ;PARAMETERS
                        ;MAINTENANCE MODE FOR THIS DEVICE
```

```
   681  002000  177777                  DZ.END: 177777
```

```
                        ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
                        ;POINTERS TO SUBROUTINES CAN BE FOUND
                        ;IN THE TABLE IMMEDIATELY FOLLOWING THE DEFINITIONS

                        ;:*****************************************************************
                        ;-----------------------------------------------------------------
002002                  .TRPTAB:
104400                  ADVANCE=TRAP+0          ;CALL TO ADVANCE TO NEXT TEST( OR SCOPE THIS ONE)
002002  006724                  .ADVANCE
104401                  SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
002004  006236                  .SCOP1
104402                  TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
002006  005262                  .TYPE
104403                  INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROUTINE
002010  00603C                  .INSTR
104404                  INSTER=TRAP+4           ;CALL TO INPUT ERROR HANDLER
002012  006134                  .INSTER
104405                  PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
002014  006154                  .PARAM
104406                  SETFLG=TRAP+6           ;CALL TO  SET FLAG ROUTINE
002016  010500                  .SETFLG
104407                  SAVOS=TRAP+7            ;CALL TO REGISTER SAVE ROUTINE
002020  006354                  .SAVOS
104410                  RESOS=TRAP+10           ;CALL TO REGISTER RESTORE ROUTINE
002022  006414                  .RESOS
104411                  CONVRT=TRAP+11          ;CALL TO DATA OUTPUT ROUTINE
002024  006446                  .CONVRT
104412                  CNVRT=TRAP+12           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
002026  006452                  .CNVRT
104413                  DEVICE.CLR=TRAP+13              ;CALL TO ISSUE A DEVICE CLEAR
002030  006652                  .DEVICE.CLR
104414                  DELAY=TRAP+14           ;CALL TO DELAY FOR FAST CPU'S
002032  006704                  .DELAY
104415                  PARMD=TRAP+15           ;CONVERT DECIMAL STRING TO OCTAL
002034  025516                  .PARMD
104416                  PAWCH=TRAP+16           ;SET FLAG    ECHO OR  CABLE
002036  025712                  .PAWCH
104417                  DCLASM=TRAP+17          ;CLEAR DEVICE, SET MAINT. BIT IF I MODE
002040  006672                  .DCLASM

                        ;-----------------------------------------------------------------
                        ;:*****************************************************************
```

```
                                        ;DZ11 VECTOR AND REGISTER INDIRECT POINTERS
                                        ;WORKING AREA
                                        
002042  160040          DZCSR:  160040  ;R/W
002044  160041          HDZCSR: 160041  ;R/W
002046  160042          DZRBUF: 160042  ;READ ONLY
002050  160043          HDZRBUF:        160043  ;READ ONLY
002052  160042          DZLPR:  160042  ;WRITE ONLY
002054  160043          HDZLPR: 160043  ;WRITE ONLY
002056  160044          DZTCR:  160044  ;R/W
002060  160045          HDZTCR: 160045  ;R/W
002062  160046          DZMSR:  160046  ;READ ONLY
002064  160047          HDZMSR: 160047  ;READ ONLY
002066  160046          DZTDR:  160046  ;WRITE ONLY
002070  160047          HDZTDR: 160047  ;WRITE ONLY
                                ;DEFAULT DZ VECTORS
002072  000300          DZRIV:  300     ;REC INTR VECTOR
002074  000302          DZRIS:  302     ;REC INTR STATUS
002076  000304          DZTIV:  304     ;XMIT INTR VECTOR
002100  000306          DZTIS:  306     ;XMIT INTR STATUS
```

```
                                        ;TIME TABLE FOR RELATIVE TIMING TESTS
                                        ;-----------------------------------

002102                                  TMTBL:
002102  000000                          T50:    0
002104  000000                          T75:    0
002106  000000                          T110:   0
002110  000000                          T134:   0
002112  000000                          T150:   0
002114  000000                          T300:   0
002116  000000                          T600:   0
002120  000000                          T1200:  0
002122  000000                          T1800:  0
002124  000000                          T2000:  0
002126  000000                          T2400:  0
002130  000000                          T3600:  0
002132  000000                          T4800:  0
002134  000000                          T7200:  0
002136  000000                          T9600:  0
002140  000000                          TEIGHT: 0
002142  000000                          TSEVEN: 0
002144  000000                          TSIX:   0
002146  000000                          TFIVE:  0
```

```
                                        ;PROGRAM INITIALIZATION
                                        ;LOCK OUT INTERRUPTS
                                        ;SET UP PROCESSOR STACK
                                        ;SET UP POWER FAIL VECTOR
                                        ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
                                        ;TYPE TITLE MESSAGE

002150                          .START:
002150  000005                  RESET                   ;CLEAR THE WORLD. START NEW ENVIRONMENT
002152  012706  001120          MOV     #STACK,SP       ;SET UP STACK
002156  106427  000340          MTPS    #PR7            ;LOCK OUT INTERRUPTS
002162  012737  007646  000024  MOV     #$PWRDN,@#24    ;SET UP POWER FAIL VECTOR
002170  113737  001410  001411  MOVB    DZNUM,SAVNUM    ;SAVE NUMBER OF DEVICES IN SYSTEM.
002176  005037  001242          CLR     $PASS           ;CLEAR PASS COUNT
002202  105037  001123          CLRB    $ERFLG          ;CLEAR ERROR FLAG
002206  012737  001500  001412  MOV     #DZ.MAP,ACTIVE  ;GET MAP POINTER.
002214  012737  000001  001406  MOV     #1,RUN          ;POINT POINTER TO FIRST DEVICE.
002222  005037  001132          CLR     $ERTTL          ;CLEAR ERROR COUNT
002226  005037  001136          CLR     $ERRPC          ;CLEAR LAST ERROR POINTER
002232  005037  001122          CLR     $TSTNM          ;SET UP FOR TEST 1
002236  012737  002150  001126  MOV     #.START,$LPADR  ;SET UP FOR POWER FAIL BEFORE
                                                        ;TESTING STARTS
                                ;SET UP FOR SMALL 11 SWITCH REGISTER COMPATIBILITY
002244  013746  000006          MOV     6,-(SP)         ;SAVE BUS ERROR PS
002250  013746  000004          MOV     4,-(SP)         ;SAVE BUS ERROR PC
002254  012737  002274  000004  MOV     #20$,4          ;SET UP TO TRAP TO THIS ROUTINE
002262  022777  177777  176670  CMP     #-1,@SWR        ;CAN 177570 BE REFERENCED?
002270  001402                  BEQ     22$             ;IF SO AND IT IS -1, TREAT LIKE SWITCHLESS
002272  000407                  BR      21$             ;IF YES,SKIP AROUND THE SETUP
002274  022626          20$:    POP2SP                  ;REMOVE THE TRAP FROM THE STACK
002276  012737  000176  001160  22$:    MOV     #SWREG,SWR      ;IF NO TRAP COMES HERE.POINT TO SOFTWARE SWR
002304  012737  000174  001162          MOV     #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY REGISTER
002312  012637  000004          21$:    MOV     (SP)+,4         ;RESTORE THE BUS ERROR VECTOR
002316  012637  000006          MOV     (SP)+,6
002322  105737  001415          TSTB    INIFLG          ;TITLE ALREADY PRINTED?
002326  001010                  BNE     29$             ;BRANCH IF YES
002330  023727  000042  004726  CMP     @#42,#$ENDAD    ;RUNNING UNDER ACT?
002336  001402                  BEQ     31$             ;IF YES DONT PRINT TITLE
002340  104402  001000          TYPE    ,MTITLE         ;PRINT THE DIAGNOSTIC'S TITLE
002344  105337  001415          31$:    DECB    INIFLG          ;SET THE ONCE ONLY FLAG
002350  105737  001255          29$:    TSTB    $ENVM           ;DETERMINE WHETHER APT SIZING SHOULD BE DONE
002354  100006                  BPL     30$             ;IF NOT, GO CHECK FOR AUTO-SIZING
002356  004737  011440          JSR     PC,SETAPT       ;OTHERWISE, GO DO APT SIZING FROM ETABLE
002362  105037  001416          CLRB    HDRFLG          ;MAKE SURE STATUS TABLE IS PRINTED
002366  000137  004270          JMP     16$             ;GO PRINT DZ STATUS TABLE
002372  032777  000001  176560  30$:    BIT     #SW00,@SWR      ;RESELECT ?
002400  001011                  BNE     32$             ;IF YES, GO SET UP THE INFORMATION
002402  122737  000377  001415  CMPB    #377,INIFLG     ;ON 1ST START; MUST ANSWER QUESTION
002410  001003                  BNE     .+10            ;IF NOT ANSWERING QUESTIONS
002412  105777  176542          TSTB    @SWR            ;ARE U AUTO SIZING?
002416  100402                  BMI     32$             ;NO AUTO SIZE! NO SW00=1 ON 1ST START!
002420  000137  003114          JMP     73$             ;IF NO, SKIP THE INTERROGATION
002424  012700  001500          32$:    MOV     #DZ.MAP,R0      ;POINT TO THE BEGINNING OF THE MAP TABLE
002430  105037  001416          CLRB    HDRFLG          ;MAKE SURE A MAP GETS PRINTED
002434  005020                  65$:    CLR     (R0)+           ;CLEAR A TABLE LOCATION
```

```
                        826  002436  020027  002000                  CMP     R0,#DZ.END      ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
                        827  002442  001374                          BNE     65$             ;IF NOT ,CLEAR THE NEXT LOCATION IN THE TABLE
                        828  002444  105337  001415                  DECB    INIFLG          ;INSURE NO AUTO SIZING IF QUESTIONS ANSWERED!
                        829
                        830                                  ;THE FOLLOWING ARE PARAMETERS USED TO FILL IN THE MAP
                        831                                  ;TABLE AND SET UP THE DIAGNOSTIC.
                        832
                        833                                          ;GET THE BASE ADDRESS OF THE DZ11'S
                        834
                        835  002450                          33$:
                        836  002450  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
                        837  002452  003334                          66$                     ;POINTER TO MESSAGE TO BE PRINTED
                        838  002454  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
                        839  002456  160000                          160000                  ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                        840  002460  163770                          163770                  ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                        841  002462  001500                          DZCR0                   ;POINTER TO MAP LOCATION TO BE FILLED
                        842  002464     007                          .BYTE   7               ;MASK OF INVALID BITS FOR THIS PARAMETER
                        843  002465     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
                        844  002466  013737  001500  001310          MOV     DZCR0,$BASE     ;COPY BASE ADDRESS TO ETABLE
                        845
                        846                                          ;GET THE BASE VECTOR ADDRESS
                        847
                        848  002474                          34$:
                        849  002474  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
                        850  002476  003400                          67$                     ;POINTER TO MESSAGE TO BE PRINTED
                        851  002500  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
                        852  002502  000300                          300                     ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                        853  002504  000776                          776                     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                        854  002506  001502                          DZVC0                   ;POINTER TO MAP LOCATION TO BE FILLED
                        855  002510     003                          .BYTE   3               ;MASK OF INVALID BITS FOR THIS PARAMETER
                        856  002511     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
                        857  002512  013737  001502  001304          MOV     DZVC0,$VECT1    ;COPY VECTOR TO ETABLE
                        858
                        859                                          ;GET THE BUS REQUEST LEVEL
                        860
                        861  002520  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
                        862  002522  003441                          68$                     ;POINTER TO MESSAGE TO BE PRINTED
                        863  002524  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
                        864  002526  000004                          4                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                        865  002530  000007                          7                       ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                        866  002532  001504                          DZLV0                   ;POINTER TO MAP LOCATION TO BE FILLED
                        867  002534     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
                        868  002535     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
                        869  002536  113737  001504  001305          MOVB    DZLV0,$VECT1+1  ;GET BUS REQUEST LEVEL INTO ETABLE
                        870  002544  106337  001305                  ASLB    $VECT1+1        ;ALIGN THE BITS PROPERLY
                        871  002550  106337  001305                  ASLB    $VECT1+1        ;ALIGN THE BITS PROPERLY
                        872  002554  106337  001305                  ASLB    $VECT1+1        ;ALIGN THE BITS PROPERLY
                        873  002560  106337  001305                  ASLB    $VECT1+1        ;ALIGN THE BITS PROPERLY
                        874  002564  106337  001305                  ASLB    $VECT1+1        ;ALIGN THE BITS PROPERLY
                        875
                        876                                          ;FIND OUT IF MODULE IS EIA OR 20 MA.
                        877
                        878  002570  104402  004130                  TYPE    ,74$            ;PRINT EIA MESSAGE
                        879  002574  005037  001220                  CLR     $TMP1           ;USE $TMP1
                        880  002600  105777  176360          80$:    TSTB    @$TKS           ;IS KEYBOARD DONE?
                        881  002604  100375                          BPL     80$             ;IF NOT, WAIT FOR IT
```

```
882  002606  017746  176354                 MOV     @$TKB,-(SP)     ;IF YES, PUT CHARACETR ON STACK
883  002612  042716  000240                 BIC     #240,(SP)       ;STRIP DOWN CHARACTER
884  002616  122726  000015                 CMPB    #15,(SP)+       ;IS IT ?
885  002622  001414                         BEQ     81$             ;IF SO, GET OUT
886  002624  014677  176342                 MOV     -(SP),@$TPB     ;IF NOT, PRINT CHARACTER
887  002630  042737  100000  001504         BIC     #BIT15,DZLVO    ;CLEAR EIA FLAG
888  002636  122726  000102                 CMPB    #102,(SP)+      ;IS IT A B?
889  002642  001356                         BNE     80$             ;IF NOT, GO BACK FOR INPUT
890  002644  052737  100000  001504         BIS     #BIT15,DZLVO    ;IF SO, SET FLAG
891  002652  000752                         BR      80$             ;GET MORE INPUT
892  002654                          81$:
893
894                                          ;GET THE MODE OF OPERATION (E,I,S)
895
896  002654  104403                         INSTR                   ;CALL THE STRING INPUT ROUTINE
897  002656  003652                         72$                     ;POINTER TO THE MESSAGE TO BE PRINTED
898  002660  104406                         SETFLG                  ;CALL THE MAINTENANCE FLAG SETUP ROUTINE
899  002662  001512                         MANTO                   ;THIS IS THE FLAG BEING SETUP
900
901                                          ;GET THE NUMBER OF DZ11'S RUNNING
902
903  002664  104403                         INSTR                   ;CALL THE STRING INPUT ROUTINE
904  002666  003610                         71$                     ;POINTER TO MESSAGE TO BE PRINTED
905  002670  104405                         PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
906  002672  000001                         1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
907  002674  000020                         16.                     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
908  002676  001220                         $TMP1                   ;POINTER TO MAP LOCATION TO BE FILLED
909  002700     000                         .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
910  002701     001                         .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
911
912  002702  012737  000377  001506         MOV     #377,LINEO      ;SET UP DEFAULT LINES
913  002710  012737  017470  001510         MOV     #17470,PARO     ;SET UP DEFAULT LPR PARAMETER
914                                          ;RECEIVER ON; 19.2 KBAUD; 2STOP BITS; 8 BIT/CHAR
915  002716  012737  000001  006722         MOV     #1,DLYCNT       ;INITIALIZE DELAY COUNT
916  002724  032777  000010  176226         BIT     #SW03,@SWR      ;DO YOU WANT PARAMETERS?
917  002732  001402                         BEQ     40$             ;IF NO, SKIP THE PARAMETER CALL
918  002734  004737  003144                 JSR     PC,23$          ;GET PARAMETERS
919  002740  012737  000001  001312  40$:   MOV     #1,$DEVM        ;INITIALIZE ACTIVE DEVICE SELECTION PARAMETER
920  002746  113737  001220  001410         MOVB    $TMP1,DZNUM     ;COPY THE NUMBER OF DEVICES
921  002754  113737  001220  001411         MOVB    $TMP1,SAVNUM    ;COPY A BACKUP NUMBER
922  002762  005337  001220          62$:   DEC     $TMP1           ;$TMP1 CONTAINS THE COUNT OF UNINITIALIZED
923  002766  001404                         BEQ     61$             ; SELECTED DEVICES
924  002770  000261                         SEC                     ;SET A BIT FLAG TO INDICATE AN ACTIVE DEVICE
925  002772  006137  001312                 ROL     $DEVM           ;POINT TO THE NEXT DEVICE
926  002776  000771                         BR      62$             ;GO DO THIS PROCEDURE AGAIN
927  003000  013737  001312  001222  61$:   MOV     $DEVM,$TMP2     ;# OF TIMES
928  003006  013737  001312  001404         MOV     $DEVM,DZACTV    ;COPY THE ACTIVE DEVICE PARAMETER
929  003014  012700  001500                 MOV     #DZCR0,R0       ;SET A POINTER TO THE SPECIFIED INFORMATION
930  003020  012701  001514                 MOV     #DZCR1,R1       ;POINT R1 TO THE REST OF THE MAP TABLE
931  003024  012702  001320                 MOV     #$DDW0,R2       ;POINT TO ETABLE'S DEVICE DESCRIPTOR WORDS
932  003030  000241                         CLC                     ;INITIALIZE THE "C" BIT FOR A ROTATION
933  003032  006037  001222                 ROR     $TMP2           ;SKIP MAPPING SETUP FOR DEVICE 0- IT'S DONE
934  003036  006237  001222          64$:   ASR     $TMP2           ;ISOLATE A SELECTION FLAG IN THE "C" BIT
935  003042  103404                         BCS     41$             ;IS THIS DEVICE SELECTED? IF YES, GO LOAD TABLE
936  003044  012711  177777                 MOV     #-1,(R1)        ;TERMINATE THE LIST
937  003050  000137  004244                 JMP     63$             ;GO TO THE NEXT BLOCK
```

```
                                41$:    MOV     (R0)+,(R1)      ;ADDRESS
                                        ADD     #10,(R1)+       ;POINT TO THE NEXT DZ11 ADDRESS VALUE
                                        MOV     (R0)+,(P1)      ;VECTOR
                                        ADD     #10,(R1)+       ;POINT TO THE NEXT VECTOR VALUE
                                        MOV     (R0)+,(R1)+     ;LE,EL
                                        MOV     (R0)+,(R1)+     ;LINES
                                        MOV     -4(R0),(R2)     ;GET THE EIA FLAG FROM THE PRIORITY WORD
                                        BIC     #77777,(R2)     ;ISOLATE THAT FLAG
                                        BIS     (R0),(R2)+      ;ADD PARAMETERS TO DEVICE DESCRIPTOR WORD
                                        MOV     (R0)+,(R1)+     ;PARAMETERS
                                        MOV     (R0)+,(R1)+     ;MAINTENANCE MODE
                                        BR      64$
003114  032777  000010  176036  73$:    BIT     #SW03,@SWR      ;ASK PARAMETERS ?
003122  001002                          BNE     42$             ;IF NO, GO DO AUTO SIZING
003124  000137  004244                  JMP     63$             ;GO SET UP FOR AUTO SIZING
003130  004737  003144          42$:    JSR     PC,23$          ;GO ASK PARAMETERS
003134  105337  001415                  DECB    INIFLG          ;INSURE NO AUTO SIZE IF QUESTIONS ANSWERED
003140  000137  004270                  JMP     16$             ;GO TO THE NEXT BLOCK

                                ;GET THE ACTIVE LINES PARAMETER

003144                          23$:
003144  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
003146  003464                          69$                     ;POINTER TO MESSAGE TO BE PRINTED
003150  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
003152  000001                          1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
003154  000377                          377                     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
003156  001506                          LINE0                   ;POINTER TO MAP LOCATION TO BE FILLED
003160     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
003161     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
003162  105037  001416                  CLRB    HDRFLG          ;MAKE SURE THE CHANGES ARE PRINTED

                                ;THIS SEGMENT CHECKS TO MAKE SURE THE LINE PARAMETER JUST ENTERED
                                ;IS LEGITIMATE IN STAGGERED MODE OPERATION IF THAT MODE WAS SELECTED

003166  005737  001512                  TST     MANT0           ;IS STAGGERED THE MODE OF OPERATION?
003172  100021                          BPL     26$             ;IF NOT, SKIP THIS SEGMENT
003174  013703  001506                  MOV     LINE0,R3        ;GET A SCRATCH COPY OF THE ACTIVE LINES
003200  006003                  24$:    ROR     R3              ;GET A LINE SELECTION BIT(EVEN NUMBER LINE)
003202  103410                          BCS     25$             ;IF IT IS SELECTED, CHECK TO SEE IF THE NEXT IS TOO
003204  001414                          BEQ     26$             ;IF ALL HAVE BEEN CHECKED, CONTINUE PROCESSING
003206  006203                          ASR     R3              ;IF IT IS 0,CHECK TO SEE IF THE NEXT IS TOO
003210  103373                          BCC     24$             ;IF THIS ONE'S 0 TOO, GO CHECK THE NEXT PAIR
003212  104402  001230          27$:    TYPE    ,$QUES          ;THIS IS AN INCORRECT PARAMETER
003216  104402  010424                  TYPE    ,MBADLN         ;LET THE USER KNOW ABOUT IT
003222  000750                          BR      23$             ;GO GET THE CORRECT PARAMETER
003224  001772                  25$:    BEQ     27$             ;IF ANOTHER FLAG ISN'T SET, THERE'S AN ERROR
003226  006203                          ASR     R3              ;GET THE NEXT FLAG
003230  103370                          BCC     27$             ;IF IT ISN'T SET, THERE'S AN ERROR
003232  000241                          CLC                     ;INITIALIZE THE "C" BIT FOR TESTING OF THE NEXT PAIR
003234  000761                          BR      24$             ;GO TEST THE NEXT PAIR OF FLAGS

                                ;GET THE LINE PARAMETER REGISTER ARGUMENT

003236                          26$:
003236  104402                          INSTR                   ;CALL THE STRING INPUT ROUTINE
```

```
                                                    70$                     ;POINTER TO MESSAGE TO BE PRINTED
                                                    PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
                                                    0                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                                                    17                      ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                                                    PAR0                    ;POINTER TO MAP LOCATION TO BE FILLED
                                                    .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
                                                    .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
                                                    MOV     #LINE0,R2       ;POINT TO THE LINE SELECTION PARAMETER
                                                    MOV     #PAR0,R3        ;POINT TO THE CHOSEN PARAMETERS
                                                    MOV     (R3),R4         ;USE BAUD RATE AS AN INDEX IN DELAY TABLE
                                                    ASL     R4              ;ALIGN INDEX ON WORD BOUNDARY
                                                    MOV     DLYTBL(R4),DLYCNT ;SET THE DELAY COUNT FOR THIS BAUD RATE
                                                    SWAB    (R3)            ;PLACE IN HIGH BYTE
                                                    BIS     #10070,(R3)     ;PLACE EXTRA PARAMETERS INTO LOC
                                            28$:    MOV     (R2),14(R2)     ;LOAD THE LINES
                                                    MOV     (R3),14(R3)     ;LOAD THE PARAMETERS
                                                    ADD     #14,R2          ;POINT TO THE NEXT SET
                                                    ADD     #14,R3          ;   OF BOTH PARAMETERS
                                                    CMP     R3,#PAR17       ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
                                                    BNE     28$             ;IF NOT, GO LOAD SOME MORE PARAMETERS
                                                    RTS     PC              ;RETURN TO CALLING BLOCK
                                            66$:    .ASCIZ  <200>/1ST CSR ADDRESS (160000:163700):  /
                                            67$:    .ASCIZ  <200>/1ST VECTOR ADDRESS (300:770):  /
                                            68$:    .ASCIZ  <200>/BR LEVEL (4:6):  /
                                            69$:    .ASCIZ  <200>/LINES ACTIVE BY BIT <IN OCTAL>(001:377):  /
                                            70$:    .ASCIZ  <200>/DEFAULT BAUD RATE <IN OCTAL>(00:17):  /
                                            71$:    .ASCIZ  <200>/# OF DZ11'S <IN OCTAL> (1:20):  /
                                            72$:    .ASCII  <200>/MAINTENANCE MODE/
                                                    .ASCII  <200>/ [EXTERNAL  <H325>-EIA ONLY       (E)]/
                                                    .ASCII  <200>/ [INTERNAL  <DZCSR03=1>           (I)]/
                                                    .ASCII  <200>/ [STAGGERED <H3271>-EIA ONLY      (S)]: /
                                                    .ASCIZ  <200>/ [STAGGERED <H3190>-20MA ONLY     (S)]: /
                                            74$:    .ASCIZ  <200>/TYPE "A" FOR EIA MODULE OR "B" FOR 20 MA (A:B): /
                                            75$:    .ASCIZ  <200>/ENTER DELAY PARAMETER: /
                                                    .EVEN
                                            63$:
                                                    CMPB    #377,INIFLG     ;ONLY DO AUTO SIZE ON 1ST START
                                                    BNE     16$             ;
                                                    BIT     #BIT7,@SWR      ;BIT7=1??
                                                    BNE     16$             ;BR IF NO AUTO SIZE
                                                    JSR     PC,AUTO.SIZE    ;GO DO THE AUTO SIZE
                                            16$:    TSTB    HDRFLG          ;HAS THE TABLE BEEN TYPED YET?
                                                    BNE     1$              ;IF SO, DON'T TYPE IT AGAIN
                                                    DECB    HDRFLG          ;INDICATE THAT THE TABLE WILL BE TYPED
                                                    TYPE    ,XHEAD          ;TYPE MAP HEADER
                                                    MOV     #DZ.MAP,R0      ;SET POINTER
                                            5$:     MOV     R0,$TMP1        ;POINT TO THE MAP LOCATION
                                                    MOV     (R0)+,$TMP2     ;SET DATA
                                                    CMP     #-1,$TMP2       ;END OF LIST?
                                                    BEQ     1$              ;BR IF YES
                                            17$:    CONVRT                  ;CALL THE OCTAL TO ASCII CONVERSION ROUTINE
                                                    XSTATQ                  ;CONVERT THE DATA AT THIS ADDRESS
                                                    BR      5$              ;GO PRINT THE NEXT PARAMETER
                                            1$:     TST     @#42            ;IS PROGRAM RUNNING UNDER MONITOR
                                                    BNE     3$              ;YES
                                                    BIT     #SW06,@SWR      ;DESELECT SPECIFIC DEVICES??
```

```
1036  004354  001422                        BEQ     3$              ;BR IF NO.
1037  004356  104402  010320                TYPE    ,MNEW           ;TYPE THE MESSAGE.
1038  004362  005000                        CLR     R0              ;ZERO DATA DISPLAY
1039  004364  000000                        HALT                    ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
1040  004366  027737  174566  001312        CMP     @SWR,$DEVM      ;IS THE NUMBER VALID?
1041  004374  101404                        BLOS    2$              ;BR IF NUMBER IS OK.
1042  004376  104402  010172                TYPE    ,MERR3          ;TELL USER OF INVALID NUMBER.
1043  004402  000000                9$:     HALT                    ;STOP EVERY THING.
1044  004404  000776                        BR      9$              ;RESTART THE PROGRAM AGAIN.
1045  004406  017737  174546  001404  2$:   MOV     @SWR,DZACTV     ;GET NEW DEVICE PATTERN
1046  004414  013700  001404                MOV     DZACTV,R0       ;SHOW THE USER WHAT HE SELECTED.
1047  004420  000000                        HALT                    ;CONTINUE DYNAMIC SWITCHES.
1048  004422  032777  000020  174530  3$:   BIT     #SW04,@SWR      ;CHECK TO SEE IF DELAY COUNT CHANGES
1049  004430  001407                        BEQ     18$             ;IF NOT, GO CLEAR VECTOR AREA
1050  004432  104403                        INSTR                   ;CALL THE STRING INPUT ROUTINE
1051  004434  004212                        75$                     ;POINTER TO MESSAGE TO BE PRINTED
1052  004436  104405                        PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1053  004440  000001                        1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1054  004442  177777                        177777                  ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1055  004444  006722                        DLYCNT                  ;POINTER TO MAP LOCATION TO BE FILLED
1056  004446     000                        .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
1057  004447     001                        .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
1058  004450  012700  000300        18$:    MOV     #300,R0         ;PREPARE TO CLEAR THE FLOATING
1059  004454  012701  000302                MOV     #302,R1         ;VECTOR AREA. 300-776
1060  004460  010120                4$:     MOV     R1,(R0)+        ;START PUTTING "PC+2 - HALT"
1061  004462  005021                        CLR     (R1)+           ;IN VECTOR AREA.
1062  004464  022021                        CMP     (R0)+,(R1)+     ;POP POINTERS
1063  004466  022700  001000                CMP     #1000,R0        ;ALL DONE??
1064  004472  001372                        BNE     4$              ;BR IF NO.
1065
1066                                        ;TEST START AND RESTART
1067                                        ;----------------------
1068
1069  004474  012706  001120        .BEGIN: MOV     #STACK,SP       ;SET UP STACK
1070  004500  106427  000340                MTPS    #PR7            ;LOCK OUT INTERRUPTS
1071  004504  005737  000042                TST     @#42            ;IS PROGRAM UNDER MONITOR CONTROL
1072  004510  001015                        BNE     2$              ;BR IF YES
1073  004512  032777  000004  174440        BIT     #BIT2,@SWR      ;CHECK FOR LOCK ON TEST
1074  004520  001406                        BEQ     1$              ;BR IF NO LOCK DESIRED.
1075  004522  104402  010216                TYPE    ,MLOCK          ;TYPE LOCK SELECTED.
1076  004526  012737  000240  005010        MOV     #NOP,TTST       ;ADJUST SCOPE ROUTINE.
1077  004534  000403                        BR      2$              ;CONTINUE ALONG.
1078  004536  013737  005232  005010  1$:   MOV     BRW,TTST        ;PREPARE NORMAL SCOPE ROUTINE
1079  004544  012737  011070  001126  2$:   MOV     #CYCLE,$LPADR   ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
1080  004552  104402  010107                TYPE    ,MR             ;TYPE "RUNNING"
1081  004556  000177  174344                JMP     @$LPADR         ;START TESTING
```

```
                                        ;END OF PASS
                                        ;TYPE NAME OF TEST
                                        ;UPDATE PASS COUNT
                                        ;CHECK FOR EXIT TO ACT-11
                                        ;RESTART TEST
                                .SBTTL  END OF PASS ROUTINE

                                ;;*****************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO CYCLE

004562                          $EOP:
                                        SCOPE
004562  000004
004564  005037  001136                  CLR     $ERRPC          ;CLEAR LAST ERROR PC
004570  105037  001123                  CLRB    $ERFLG          ;CLEAR ERROR FLAG
004574  104402  010063                  TYPE    ,MEPASS         ;TYPE END PASS
004600  104402  010245                  TYPE    ,MCSRX          ;TYPE CSR
004604  104412  004742                  CNVRT   ,XCSR           ;SHOW IT
004610  104402  010253                  TYPE    ,MVECX          ;TYPE VECTOR
004614  104412  004750                  CNVRT   ,XVEC           ;SHOW IT
004620  005237  001242                  INC     $PASS           ;RAISE PASS COUNT
004624  104402  010261                  TYPE    ,MPASSX         ;TYPE PASSES
004630  104412  004756                  CNVRT   ,XPASS          ;SHOW IT
004634  005337  001242                  DEC     $PASS           ;RESTORE PASS COUNT
004640  104402  010272                  TYPE    ,MERRX          ;TYPE ERRORS
004644  104412  004764                  CNVRT   ,XERR           ;SHOW IT
004650  105337  001411                  DECB    $AVNUM          ;ARE ALL DEVICES TESTED?
004654  001030                          BNE     $DOAGN          ;BR IF NO.
004656  113737  001410  001411          MOVB    DZNUM,SAVNUM    ;RESTORE THE COUNT
004664  005037  001226                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
004670  005237  001242                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
004674  042737  100000  001242          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
004702  005327                          DEC     (PC)+           ;;LOOP?
004704  000001                  $EOPCT: .WORD   1
004706  003013                          BGT     $DOAGN          ;;YES
004710  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
004712  000001                  $ENDCT: .WORD   1
004714  004704                          $EOPCT
004716  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
004722  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
004724  000005                          RESET                   ;;CLEAR THE WORLD
004726  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
004730  000240                          NOP                     ;;SAVE ROOM
004732  000240                          NOP                     ;;FOR
004734  000240                          NOP                     ;;ACT11
004736                          $DOAGN:
004736  000137                          JMP     @(PC)+          ;;RETURN
004740  011070                  $RTNAD: .WORD   CYCLE
                                ;
004742  000001                  XCSR:   1
004744     006      002                 .BYTE   6,2
004746  002042                          DZCSR
004750  000001                  XVEC:   1
004752     003      002                 .BYTE   3,2
004754  002072                          DZRIV
```

```
                                XPASS:  1
                                        .BYTE   6,2
                                        $PASS
                                XERR:   1
                                        .BYTE   6,2
                                        $ERTTL

                                        ;SCOPE LOOP AND ITERATION HANDLER
                                        ;-------------------------------

                                .SBTTL  SCOPE HANDLER ROUTINE

                                ;;*****************************************************************
                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW14=1         LOOP ON TEST
                                ;*SW11=1         INHIBIT ITERATIONS
                                ;*CALL
                                ;*      SCOPE           ;;SCOPE=IOT

004772                          $SCOPE:
004772  004737  007360          .SCOPE: JSR     PC,SERV.G       ;FIND OUT IF <↑G> WAS HIT
004776  005037  001136                  CLR     $ERRPC          ;CLEAR LAST ERROR PC.
005002  022716  012376                  CMP     #TST1+2,(SP)    ;IS THIS THE SCOPE AT THE BEGINNING OF TST1?
005006  001413                          BEQ     $XTSTR          ;IF SO, DON'T LOOP ON IT
005010  000406                  TTST:   BR      1$              ;GOTO 1$    (IF LOCK SW02=1; THIS LOC =240)
005012  105777  174146                  TSTB    @$TKS           ;KEYBOARD DONE?
005016  100067                          BPL     $OVER           ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
005020  017766  174142  177776          MOV     @$TKB,-2(SP)    ;CLEAR DONE BIT
005026  032777  040000  174124  1$:     BIT     #BIT14,@$WR     ;;LOOP ON PRESENT TEST?
005034  001060                          BNE     $OVER           ;;YES IF SW14=1
                                ;*****START OF CODE FOR THE XOR TESTER*****
005036  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
005040  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
005044  012737  005064  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
005052  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
005056  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
005062  000436                          BR      $SVLAD          ;;GO TO THE NEXT TEST
005064  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
005066  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
005072  000441                          BR      $OVER           ;;LOOP ON THE PRESENT TEST
005074                          6$:;*****END OF CODE FOR THE XOR TESTER*****
005074  105737  001123          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
005100  001404                          BEQ     3$              ;;BR IF NO
005102  105037  001123          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
005106  005037  001226                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
005112  032777  004000  174040  3$:     BIT     #BIT11,@$WR     ;;INHIBIT ITERATIONS?
005120  001011                          BNE     1$              ;;BR IF YES
005122  005737  001242                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
005126  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
005130  005237  001124                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
005134  023737  001226  001124          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
005142  002015                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
```

```
1194  005144  012737  000001  001124  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
1195  005152  013737  005234  001226          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
1196  005160  105237  001122          $SVLAD: INCB    $ISTNM          ;;COUNT TEST NUMBERS
1197  005164  113737  001122  001240          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
1198  005172  011637  001126                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
1199  005176  013777  001122  173756  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
1200  005204  013716  001126                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
1201  005210  105037  001417          3$:     CLRB    MNTFLG          ;CLEAR THE MAINTENANCE BIT SETTER AFTER EACH TEST
1202  005214  005737  001370                  TST     MODE            ;HAS THE MODE BEEN CHANGED?
1203  005220  001003                          BNE     4$              ;IF NOT INTERNAL , GO DO A TEST
1204  005222  112737  000010  001417          MOVB    #MAINT,MNTFLG   ;IF INTERNAL MODE NOW, SET THE MAINTENANCE BIT
1205  005230  000002          4$:     RTI                             ;GO DO THE TEST
1206  005232  000406          BRW:    406
1207  005234  000005          $MXCNT: 5                               ;;MAX. NUMBER OF ITERATIONS
1208
1209                                  ;CHECK FOR FREEZE ON CURRENT DATA
1210                                  ;-------------------------------
1211
1212  005236  032777  001000  173714  .SCOP1: BIT     #SW09,@SWR      ;IS SW09=1(SET)?
1213  005244  001405                          BEQ     1$              ;BR IF NOT SET.
1214  005246  005737  001362                  TST     LOCK            ;IS THER A TIGHT LOOP SPECIFIED?
1215  005252  001402                          BEQ     1$              ;IF NO, RETURN
1216  005254  013716  001362                  MOV     LOCK,(SP)       ;IF YES, GOTO THE ADDRESS IN LOCK.
1217  005260  000002          1$:     RTI                             ;GO BACK.
1218
1219  005262  032777  010000  173670  .TYPE:  BIT     #SW12,@SWR      ;INHIBIT ALL PRINTOUT??
1220  005270  001403                          BEQ     1$              ;IF NOT, GO TYPE
1221  005272  062716  000002                  ADD     #2,(SP)         ;SKIP OVER MESSAGE POINTER
1222  005276  000002                          RTI                     ;RETURN TO WHERE PROCEDURE WAS INVOKED
1223  005300                  1$:
1224                          .SBTTL  TYPE ROUTINE
1225
1226                          ;;**************************************************************
1227                          ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
1228                          ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
1229                          ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
1230                          ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
1231                          ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
1232                          ;*
1233                          ;*CALL:
1234                          ;*1) USING A TRAP INSTRUCTION
1235                          ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
1236                          ;*OR
1237                          ;*      TYPE
1238                          ;*      MESADR
1239                          ;*
1240
1241  005300  105737  001177          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
1242  005304  100002                          BPL     1$              ;;BR IF YES
1243  005306  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
1244  005310  000430                          BR      3$              ;;LEAVE
1245  005312  010046          1$:     MOV     R0,-(SP)        ;;SAVE R0
1246  005314  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
1247  005320  122737  000001  001254          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
1248  005326  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
1249  005330  132737  000100  001255          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
```

```
1250  005336  001405                        BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
1251  005340  010037  005350                MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
1252  005344  004737  005570                JSR     PC,$ATY2        ;;SPOOL MESSAGE TO APT
1253  005350  000000            61$:        .WORD   0               ;;MESSAGE ADDRESS
1254  005352  132737  000040  001255 62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
1255  005360  001003                        BNE     60$             ;;YES,SKIP TYPE OUT
1256  005362  112046            2$:         MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
1257  005364  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
1258  005366  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
1259  005370  012600            60$:        MOV     (SP)+,R0        ;;RESTORE R0
1260  005372  062716  000002            3$: ADD     #2,(SP)         ;;ADJUST RETURN PC
1261  005376  000002                        RTI                     ;;RETURN
1262  005400  122716  000011            4$: CMPB    #HT,(SP)        ;;BRANCH IF <HT>
1263  005404  001430                        BEQ     8$
1264  005406  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
1265  005412  001006                        BNE     5$
1266  005414  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
1267  005416  104402                        TYPE                    ;;TYPE A CR AND LF
1268  005420  001231                        $CRLF
1269  005422  105037  005556                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
1270  005426  000755                        BR      2$              ;;GET NEXT CHARACTER
1271  005430  004737  005512            5$: JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
1272  005434  123726  001176            6$: CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
1273  005440  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
1274  005442  013746  001174                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
1275                                                                ;;AND THE NULL CHAR.
1276  005446  105366  000001            7$: DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
1277  005452  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
1278  005454  004737  005512                JSR     PC,$TYPEC       ;;GO TYPE A NULL
1279  005460  105337  005556                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
1280  005464  000770                        BR      7$              ;;LOOP
1281
1282                                ;HORIZONTAL TAB PROCESSOR
1283
1284  005466  112716  000040            8$: MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
1285  005472  004737  005512            9$: JSR     PC,$TYPEC       ;;TYPE A SPACE
1286  005476  132737  000007  005556        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
1287  005504  001372                        BNE     9$              ;;TAB STOP
1288  005506  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
1289  005510  000724                        BR      2$              ;;GET NEXT CHARACTER
1290  005512  105777  173452        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
1291  005516  100375                        BPL     $TYPEC
1292  005520  116677  000002  173444        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
1293  005526  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
1294  005534  001003                        BNE     1$              ;;BRANCH IF NO
1295  005536  105037  005556                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
1296  005542  000406                        BR      $TYPEX          ;;EXIT
1297  005544  122766  000012  000002    1$: CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
1298  005552  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
1299  005554  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
1300  005556  000000        $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
1301  005560  000207        $TYPEX: RTS     PC
1302
1303                                .SBTTL  APT COMMUNICATIONS ROUTINE
1304
1305                                ;;**************************************************************
```

```
1306  005562  112737  000001  006026  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
1307  005570  112737  000001  006024  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
1308  005576  000403                  BR      $ATYC
1309  005600  112737  000001  006026  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
1310  005606                          $ATYC:
1311  005606  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
1312  005610  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
1313  005612  105737  006024                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
1314  005616  001450                          BEQ     5$              ;;IF NOT:  BR
1315  005620  122737  000001  001254          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
1316  005626  001031                          BNE     3$              ;;IF NOT:  BR
1317  005630  132737  000100  001255          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
1318  005636  001425                          BEQ     3$              ;;IF NOT:  BR
1319  005640  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
1320  005644  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
1321  005652  005737  001234          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
1322  005656  001375                          BNE     1$              ;;IF NOT:  WAIT
1323  005660  010037  001250                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
1324  005664  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
1325  005666  001376                          BNE     2$
1326  005670  163700  001250                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
1327  005674  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
1328  005676  010037  001252                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
1329  005702  012737  000004  001234          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
1330  005710  000413                          BR      5$
1331  005712  017637  000004  005736  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
1332  005720  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
1333  005726  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
1334  005732  004737  005300                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
1335  005736  000000                  4$:     .WORD   0
1336  005740                          5$:
1337  005740  105737  006026          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
1338  005744  001416                          BEQ     12$             ;;IF NOT:  BR
1339  005746  005737  001254                  TST     $ENV            ;;RUNNING UNDER APT?
1340  005752  001413                          BEQ     12$             ;;IF NOT:  BR
1341  005754  005737  001234          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
1342  005760  001375                          BNE     11$             ;;IF NOT:  WAIT
1343  005762  017637  000004  001236          MOV     @4(SF   $FATAL  ;;GET ERROR #
1344  005770  062766  000002  000004          ADD     #2,4(_  )               ;;BUMP RETURN ADDR.
1345  005776  005237  001234                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
1346  006002  105037  006026          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
1347  006006  105037  006025                  CLRB    $LFLG           ;;CLEAR LOG FLAG
1348  006012  105037  006024                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
1349  006016  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
1350  006020  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
1351  006022  000207                          RTS     PC              ;;RETURN
1352  006024     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
1353  006025     000          $LFLG:  .BYTE   0               ;;LOG FLAG
1354  006026     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
1355          006030                  .EVEN
1356          000200          APTSIZE=200
1357          000001          APTENV=001
1358          000100          APTSPOOL=100
1359          000040          APTCSUP=040
1360
1361                                  ;STRING INPUT ROUTINE
```

```
                                        ;-------------------------
                                        ;
006030  010346                  .INSTR: MOV     R3,-(SP)        ;SAVE R3 ON STACK
006032  010446                          MOV     R4,-(SP)        ;SAVE R4 ON STACK
006034  017637  000004  006052          MOV     @4(SP),.MSG     ;GET THE ADDRESS OF THE MESSAGE TO BE PRINTED
006042  062766  000002  000004          ADD     #2,4(SP)        ;POINT TO INSTRUCTION AFTER ADDRESS POINTER
006050  104402                  .INST1: TYPE                    ;PRINT THE MESSAGE
006052  000000                  .MSG:   0                       ;MESSAGE IS POINTED TO FROM HERE
006054  012704  010620                  MOV     #INBUF,R4       ;POINT R4 TO THE INPUT BUFFER
006060  012703  000007                  MOV     #7,R3           ;SET THE MAXIMUM NUMBER OF CHARACTERS ALLOWED
006064  105777  173074          1$:     TSTB    @$TKS           ;HAS A CHARACTER BEEN RECEIVED?
006070  100375                          BPL     1$              ;IF NO, KEEP WAITING FOR IT
006072  117714  173070                  MOVB    @$TKB,(R4)      ;IF YES, SAVE IT IN THE INPUT BUFFER
006076  142714  000200                  BICB    #200,(R4)       ;KEEP ONLY THE 7-BIT ASCII INFORMATION
006102  122427  000015                  CMPB    (R4)+,#15       ;IS THIS CHARACTER A LINE FEED?
006106  001417                          BEQ     INSTR2          ;IF SO, TERMINATE THE INPUT SEQUENCE
006110  105777  173054          2$:     TSTB    @$TPS           ;IF NOT, CHECK TO SEE IF THE CHARACTER CAN PRINT
006114  100375                          BPL     2$              ;IF WE CAN'T, WAIT UNTIL WE CAN
006116  017777  173044  173046          MOV     @$TKB,@$TPB     ;ECHO THE CHARACTER BACK
006124  005303                          DEC     R3              ;REDUCE THE NUMBER OF CHARACTERS RECEIVED
006126  001356                          BNE     1$              ;IF WE DON'T HAVE 7, GO GET SOME MORE
006130  012604                          MOV     (SP)+,R4        ;IF WE HAVE 7, RESTORE R4
006132  012603                          MOV     (SP)+,R3        ;RESTORE R3
006134  010346                  .INSTE: MOV     R3,-(SP)        ;SAVE R3 ON THE STACK
006136  010446                          MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
006140  104402  001230                  TYPE    ,$QUES          ;PRINT A QUESTION MARK... WHAT'S GOING ON?
006144  000741                          BR      .INST1          ;GO PRINT THE MESSAGE AGAIN
006146  012604                  INSTR2: MOV     (SP)+,R4        ;RESTORE R4
006150  012603                          MOV     (SP)+,R3        ;RESTORE R3
006152  000002                          RTI                     ;RETURN TO THE MAIN PROCEDURE

                                        ;CONVERT ASCII STRING TO OCTAL
                                        ;-----------------------------

006154  010546                  .PARAM: MOV     R5,-(SP)        ;SAVE R5 ON THE STACK
006156  010446                          MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
006160  016605  000004                  MOV     4(SP),R5        ;GET THE SETUP INFORMATION POINTER
006164  012537  006344                  MOV     (R5)+,LOLIM     ;SET THE LOW LIMIT FOR THE INPUT
006170  012537  006346                  MOV     (R5)+,HILIM     ;SET THE HIGH LIMIT FOR THE INPUT
006174  012537  006350                  MOV     (R5)+,DEVADR    ;SAVE THE ADDRESS WHERE THE RESULT WILL BE STORED
006200  112537  006352                  MOVB    (R5)+,LOBITS    ;GET THE MASK OF THE INCORRECT BITS
006204  112537  006353                  MOVB    (R5)+,ADRCNT    ;GET THE COUNT OF ITEMS TO BE STORED
006210  010566  000004                  MOV     R5,4(SP)        ;POINT TO WHERE MAIN LINE PROGRAM WILL RESUME
006214  005005                  PARAM1: CLR     R5              ;INITIALIZE THE ASCII TO OCTAL RESULT WORD
006216  012704  010620                  MOV     #INBUF,R4       ;POINT TO THE INPUT BUFFER
006222  122714  000015                  CMPB    #15,(R4)        ;IS THIS CHARACTER A CARRIAGE RETURN?
006226  001420                          BEQ     PARERR          ;IF SO, PRINT THE MESSAGE AGAIN
006230  121427  000060          1$:     CMPB    (R4),#60        ;IS THIS CHARACTER BELOW THE NUMERIC RANGE?
006234  002415                          BLT     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
006236  121427  000067                  CMPB    (R4),#67        ;IS THIS CHARACTER ABOVE THE NUMERIC RANGE?
006242  003012                          BGT     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
006244  142714  000060                  BICB    #60,(R4)        ;ISOLATE THE NUMBER THE CHARACTER REPRESENTS
006250  152405                          BISB    (R4)+,R5        ;CONCATENATE THESE BITS TO THE ALREADY EXISTING STRING
006252  122714  000015                  CMPB    #15,(R4)        ;IS THE NEXT CHARACTER A CARRIAGE RETURN?
006256  001406                          BEQ     LIMITS          ;IF SO, GO SEE IF NUMBER IS WITHIN LIMITS
006260  006305                          ASL     R5              ;CLEAR BIT POSITION 0, MOVE EXISTING STRING TO LEFT
```

```
1418  006262  006305                  ASL     R5              ;CLEAR POSITION 1, MOVE STRING TO LEFT AGAIN
1419  006264  006305                  ASL     R5              ;MOVE THE STRING ONE MORE TIME TO MAKE ROOM FOR
1420                                                          ;NEXT THREE BITS
1421  006266  000760                  BR      1$              ;GO GET THE NEXT CHARACTER
1422  006270  104404          PARERR: TNSTER                  ;THERE WAS AN ERROR... GO PRINT MESSAGE AGAIN
1423  006272  000750                  BR      PARAM1          ;TRY GETTING THE PARAMETERS AGAIN
1424
1425                                  ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
1426                                  ;--------------------------------------
1427
1428  006274  020537  006346  LIMITS: CMP     R5,HILIM        ;DOES RESULT EXCEED ITS MAXIMUM CORRECT VALUE?
1429  006300  101373                  BHI     PARERR          ;IF YES, GO PRINT THE MESSAGE AGAIN
1430  006302  020537  006344          CMP     R5,LOLIM        ;IS THE RESULT LOWER THAN ALLOWED?
1431  006306  103770                  BLO     PARERR          ;IF YES, GO PRINT THE MESSAGE AGAIN
1432  006310  133705  006352          BITB    LOBITS,R5       ;ARE ANY INCORRECT BITS SET IN THE RESULT?
1433  006314  001365                  BNE     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
1434
1435                                  ;STORE NUMBER AT SPECIFIED ADDRESS
1436
1437  006316  013704  006350          MOV     DEVADR,R4       ;POINT TO THE LOCATION WHERE THE RESULT WILL BE STORED
1438  006322  010524          1$:     MOV     R5,(R4)+        ;STORE THE RESULT
1439  006324  062705  000002          ADD     #2,R5           ;CALCULATE THE NEXT DATUM
1440  006330  105337  006353          DECB    ADRCNT          ;REDUCE COUNT OF STORED RESULTS. IS IT EXCEEDED?
1441  006334  001372                  BNE     1$              ;IF NOT, GO STORE THE NEXT DATUM
1442  006336  012604                  MOV     (SP)+,R4        ;RESTORE R4
1443  006340  012605                  MOV     (SP)+,R5        ;RESTORE R5
1444  006342  000002                  RTI                     ;RETURN TO THE MAIN PROGRAM
1445
1446  006344  000000          LOLIM:  0                       ;LOWEST ACCEPTABLE VALUE
1447  006346  000000          HILIM:  0                       ;HIGHEST ACCEPTABLE
1448  006350  000000          DEVADR: 0                       ;LOCATION WHERE RESULT WILL BE STORED
1449  006352     000          LOBITS: .BYTE   0               ;INCORRECT BITS MASK
1450  006353     000          ADRCNT: .BYTE   0               ;COUNT OF ITEMS TO BE STORED
1451
1452                                  ;SAVE PC OF TEST THAT FAILED AND R0-R5
1453                                  ;-------------------------------------
1454
1455  006354  016637  000004  001402 .SAV05: MOV     4(SP),SAVPC     ;SAVE R7 (PC)
1456
1457                                  ;SAVE R0-R5
1458
1459  006362  010537  001214          SV05:   MOV     R5,$REG5        ;SAVE R5
1460  006366  010437  001212                  MOV     R4,$REG4        ;SAVE R4
1461  006372  010337  001210                  MOV     R3,$REG3        ;SAVE R3
1462  006376  010237  001206                  MOV     R2,$REG2        ;SAVE R2
1463  006402  010137  001204                  MOV     R1,$REG1        ;SAVE R1
1464  006406  010037  001202                  MOV     R0,$REG0        ;SAVE R0
1465  006412  000002                  RTI                     ;LEAVE.
1466
1467                                  ;RESTORE R0-R5
1468
1469  006414  013700  001202          .RES05: MOV     $REG0,R0        ;RESTORE R0
1470  006420  013701  001204                  MOV     $REG1,R1        ;RESTORE R1
1471  006424  013702  001206                  MOV     $REG2,R2        ;RESTORE R2
1472  006430  013703  001210                  MOV     $REG3,R3        ;RESTORE R3
1473  006434  013704  001212                  MOV     $REG4,R4        ;RESTORE R4
```

```
1474  006440  013705  001214            MOV     $REGS,R5        ;RESTORE R5
1475  006444  000002                    RTI                     ;LEAVE
1476
1477                                    ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
1478                                    ;-----------------------------------------------------
1479
1480  006446  104402  001231    .CONVR: TYPE    ,$CRLF          ;PRINT A CARRIAGE RETURN
1481  006452  010046            .CNVRT: MOV     R0,-(SP)        ;SAVE R0
1482  006454  010146                    MOV     R1,-(SP)        ;SAVE R1
1483  006456  010346                    MOV     R3,-(SP)        ;SAVE R3
1484  006460  010446                    MOV     R4,-(SP)        ;SAVE R4
1485  006462  010546                    MOV     R5,-(SP)        ;SAVE R5
1486  006464  017601  000012            MOV     @12(SP),R1      ;PLACE THE ADDRESS OF THE ARGUMENTS IN R1
1487  006470  062766  000002  000012    ADD     #2,12(SP)       ;POINT TO WHERE MAIN PROGRAM WILL RESUME
1488  006476  012137  006622            MOV     (R1)+,WRDCNT    ;GET NUMBER OF WORDS TO BE PRINTED
1489  006502  112105            1$:     MOVB    (R1)+,R5        ;GET THE NUMBER OF CHARACTERS TO BE PRINTED
1490  006504  112100                    MOVB    (R1)+,R0        ;GET THE NUMBER OF SPACES TO PRINT
1491  006506  013104                    MOV     @(R1)+,R4       ;COPY THE WORD TO BE CONVERTED
1492  006510  110537  006624            MOVB    R5,CHRCNT       ;COPY THE CHARACTER COUNT
1493  006514  010403            3$:     MOV     R4,R3           ;COPY THE ARGUMENT WORD AGAIN
1494  006516  042703  177770            BIC     #^C<7>,R3       ;ISOLATE THREE BITS TO BE TREATED AS A CHARACTER
1495  006522  062703  000060            ADD     #060,R3         ;MAKE AN ASCII CHARACTER OUT OF THEM
1496  006526  110346                    MOVB    R3,-(SP)        ;SAVE THAT CHARACTER
1497  006530  006004                    ROR     R4              ;MOVE THE NEXT THREE BITS INTO PLACE
1498  006532  006204                    ASR     R4              ;MOVE THEM AGAIN
1499  006534  006204                    ASR     R4              ;AND FINALLY A THIRD TIME
1500  006536  005305                    DEC     R5              ;REDUCE CHARACTER COUNT.ARE ALL CHARACTERS
1501                                                            ;BUILT?
1502  006540  001365                    BNE     3$              ;IF NO, GO BUILD THE NEXT ONE.
1503  006542  012703  010724            MOV     #MDATA,R3       ;NOW POINT TO WHERE NUMBER WILL BE PRINTED FROM
1504  006546  112623            4$:     MOVB    (SP)+,(R3)+     ;STORE THE CHARACTER, STARTING WITH THE MOST
1505  006550  105337  006624            DECB    CHRCNT          ;REDUCE COUNT. ARE ALL CHARACTERS TRANSFERRED?
1506  006554  001374                    BNE     4$              ;IF NO, GO TRANSFER ANOTHER
1507  006556  105700                    TSTB    R0              ;ARE ANY SPACES TO BE PRINTED?
1508  006560  001404                    BEQ     6$              ;IF NO, DON'T SET UP ANY
1509  006562  112723  000040    5$:     MOVB    #040,(R3)+      ;ADD A SPACE TO THE OUTPUT BUFFER
1510  006566  105300                    DECB    R0              ;REDUCE THE COUNT. SHOULD WE PRINT MORE?
1511  006570  001374                    BNE     5$              ;IF YES, GO ADD ANOTHER SPACE
1512  006572  105013            6$:     CLRB    (R3)            ;TERMINATE THE OUTPUT BUFFER WITH A ZERO
1513  006574  104402  010724            TYPE    ,MDATA          ;PRINT THE STRING WE JUST BUILT
1514  006600  005337  006622            DEC     WRDCNT          ;REDUCE THE WORD COUNT. ARE ANY MORE WORDS LEFT?
1515  006604  001336                    BNE     1$              ;IF YES, GO CONVERT THEM
1516  006606  012605                    MOV     (SP)+,R5        ;RESTORE R5
1517  006610  012604                    MOV     (SP)+,R4        ;RESTORE R4
1518  006612  012603                    MOV     (SP)+,R3        ;RESTORE R3
1519  006614  012601                    MOV     (SP)+,R1        ;RESTORE R1
1520  006616  012600                    MOV     (SP)+,R0        ;RESTORE R0
1521  006620  000002                    RTI                     ;RETURN TO THE MAIN PROGRAM
1522  006622  000000            WRDCNT: 0
1523  006624     000            CHRCNT: .BYTE                   ;NUMBER OF CHARACTERS TO PRINT
1524  006625     000            SPACNT: .BYTE   0               ;NUMBER OF SPACES TO PRINT
1525
1526  006626  000000            BINWRD: 0
1527
1528
1529                                    ;TRAP DISPATCH SERVICE
```

```
                                        ;ARGUMENT OF TRAP IS EXTRACTED
                                        ;AND USED AS OFFSET TO OBTAIN POINTER
                                        ;TO SELECTED SUBROUTINE
                                        
006630  010046          .TRPSR: MOV     R0,-(SP)        ;SAVE R0. USE R0 TO FIND TRAP ROUTINE
006632  016600  000002          MOV     2(SP),R0        ;GET TRAP ADDRESS
006636  005740                  TST     -(R0)           ;GET TRAP
006640  111000                  MOVB    (R0),R0         ;GET RIGHT BYTE OF TRAP(TRAP OFFSET)
006642  006300                  ASL     R0              ;POSITION OFFSET FOR TABLE INDEXING
006644  016000  002002          MOV     .TRPTAB(R0),R0  ;PLACE INDEXED ADDRESS OF TABLE IN R0
006650  000200                  RTS     R0              ;TRANSFER TO THAT ADDRESS AND RESTORE OLD R0
                                        
                                        ;DEVICE CLEAR ROUTINE
                                        ;ISSUE A DEVICE CLEAR
                                        ;--------------------
006652                  .DEVICE.CLR:
006652  052777  000020  173162          BIS     #DCLR,@DZCSR    ;SET DCLR
006660  032777  000020  173154  1$:     BIT     #DCLR,@DZCSR    ;DID IT CLEAR?
006666  001374                  BNE     1$              ;BR IF NO
006670  000002                  RTI                     ;EXIT ROUTINE
                                        
                                        ;ROUTINE TO HANDLE MAINTENANCE BIT SETTING WITH DEVICE CLEAR
                                        ;----------------------------------------------------------
006672  104413          .DCLASM:DEVICE.CLR              ;ISSUE A DEVICE CLEAR
006674  153777  001417  173140          BISB    MNTFLG,@DZCSR   ;LOAD THE MAINTENANCE BIT IF IT IS I MODE
006702  000002                  RTI                     ;RETURN TO CALLING ROUTINE
                                        
006704                  .DELAY:
006704  010046                  MOV     R0,-(SP)        ;SAVE R0
006706  013700  006722          MOV     DLYCNT,R0       ;SET COUNT
006712  005300          1$:     DEC     R0              ;DELAY
006714  001376                  BNE     1$              ;
006716  012600                  MOV     (SP)+,R0        ;RESTORE R0
006720  000002                  RTI                     ;LEAVE ROUTINE
006722  000001          DLYCNT: .WORD     1             ;PATCHABLE LOC FOR MORE TIME
                                        
                                        ;ADVANCE TO NEXT TEST HANDLER
                                        ;----------------------------
                                        
006724  013716  001360  .ADVANCE:MOV     NEXT,(SP)       ;CRUNCH STACK WITH ADDRESS OF SCOPE CALL
006730  005037  001362          CLR     LOCK            ;RESET TIGHT LOOP ADDRESS
006734  000002                  RTI                     ;CHECK TO SEE IF OLD TEST GETS REPEATED
                                        
                                        ;ERROR HANDLER
                                        ;-------------
                                        
006736  004737  007360  $ERROR: JSR     PC,SERV.G       ;FIND OUT IF <↑G> WAS HIT
006742  032777  010000  172210          BIT     #SW12,@SWR      ;BELL ON ERROR?
006750  001406                  BEQ     XBX             ;BR IF NO BELL
006752  105777  172212          TSTB    @$TPS           ;TTY READY.
006756  100003                  BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
006760  112777  000207  172204          MOVB    #207,@$TPB      ;PUSH A BELL AT THE TTY.
006766  032777  020000  172164  XBX:    BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
006774  001113                  BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
006776  021637  001136          CMP     (SP),$ERRPC     ;WAS THIS ERROR FOUND LAST TIME?
007002  001404                  BEQ     1$              ;BR IF YES
```

```
1586  007004  011637  001136            MOV     (SP),$ERRPC     ;RECORD BFING HERE
1587  007010  105037  001123            CLRB    $ERFLG          ;PREPARE HEADER
1588  007014  104407                1$: SAVOS                   ;SAVE ALL PROC REGISTERS
1589  007016  011605                    MOV     (SP),R5         ;GET THE PC OF ERROR
1590  007020  162705  000002            SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
1591  007024  011504                    MOV     (R5),R4         ;GET ERROR INSTRUCTION
1592  007026  110437  001134            MOVB    R4,$ITEMB       ;COPY TEST NUMBER FOR APT HANDLING
1593  007032  006304                    ASL     R4              ;MULT BY TWO
1594  007034  061504                    ADD     (R5),R4         ;DOUBLE IT
1595  007036  006304                    ASL     R4              ;MULT AGAIN
1596  007040  042704  177001            BIC     #177001,R4      ;CLEAR JUNK
1597  007044  062704  027064            ADD     #.ERRTAB,R4     ;GET POINTER
1598  007050  012437  007174            MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
1599  007054  012437  007206            MOV     (R4)+,DATAHD    ;GET DATA HEADER
1600  007060  011437  007220            MOV     (R4),DATABP     ;GET DATA TABLE
1601  007064  105737  001123            TSTB    $ERFLG          ;TYPE HEADER
1602  007070  001403                    BEQ     TYPMSG          ;BR IF YES
1603  007072  005737  007220            TST     DATABP          ;DOES DATA TABLE EXIST?
1604  007076  001044                    BNE     TYPDAT          ;BR IF YES.
1605  007100  104402  001231    TYPMSG: TYPE    ,$CRLF          ;TYPE A CARRIAGE RETURN
1606  007104  104402  001231            TYPE    ,$CRLF          ;AND TYPE ANOTHER
1607  007110  005737  001362            TST     LOCK
1608  007114  001402                    BEQ     1$
1609  007116  104402  010315            TYPE    ,MASTEK
1610  007122  104402  010303        1$: TYPE    ,MTSTN
1611  007126  104412  007352            CNVRT   ,XTSTN          ;SHOW IT
1612  007132  104402  010372            TYPE    ,MERRPC         ;TYPE PC.
1613  007136  104412  007344            CNVRT   ,ERTABO         ;SHOW IT
1614  007142  104402  010245            TYPE    ,MCSRX
1615  007146  104412  004742            CNVRT   ,XCSR
1616  007152  104402  001231            TYPE    ,$CRLF          ;GIVE A CR/LF
1617  007156  112737  177777  001123    MOVB    #-1,$ERFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
1618  007164  005737  007174            TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
1619  007170  001402                    BEQ     WTBS.FM         ;BR IF NO.
1620  007172  104402                    TYPE                    ;TYPE
1621  007174  000000    ERRMSG: 0                               ;    ERROR MESSAGE
1622  007176            WTBS.FM:                                ;
1623  007176  005737  007206            TST     DATAHD          ;DATA HEADER?
1624  007202  001402                    BEQ     TYPDAT          ;BR IF NO
1625  007204  104402                    TYPE                    ;TYPE
1626  007206  000000    DATAHD: 0                               ;    DATA HEADER
1627  007210  005737  007220    TYPDAT: TST     DATABP          ;DATA TABLE?
1628  007214  001402                    BEQ     RESREG          ;BR IF NO.
1629  007216  104411                    CONVRT                  ;SHOW
1630  007220  000000    DATABP: 0                               ;    DATA TABLE
1631  007222  104410    RESREG: RESOS                           ;RESTORE PROC REGISTERS
1632  007224  122737  000001  001254    CMPB    #APTENV,$ENV    ;IS APT RUNNING?
1633  007232  001007                    BNE     2$              ;SKIP APT CALL IF NOT
1634  007234  113737  001134  007246    MOVB    $ITEMB,7$       ;COPY ERROR NUMBER
1635  007242  004737  005600            JSR     PC,$ATY4        ;CALL APT SERVICE
1636  007246  000000    7$:     .WORD   0                       ;ERROR NUMBER STUCK HERE
1637  007250  000777    8$:     BR      8$                      ;LOCK UP HERE
1638  007252  022737  004726  000042  2$:     CMP     #$ENDAD,@#42    ;CHECK TO SEE IF IN ACT-11 MODE
1639  007260  001403                    BEQ     1$              ;IF SO, HANDLE ACCORDINGLY
1640  007262  005777  171672            TST     @SWR            ;HALT ON ERROR?
1641  007266  100004                    BPL     EXITER          ;BR IF NO HALT ON ERROR
```

```
1642  007270  016677  000002  171664  1$:     MOV     2(SP),@DISPLAY  ;SHOW ERROR PC IN DATA DISPLAY
1643  007276  000000                          HALT                    ;HALT
1644  007300  005237  001132          EXITER: INC     $ERTTL          ;UPDATE ERROR COUNT
1645  007304  032777  000400  171646          BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
1646  007312  001007                          BNE     1$              ;BR IF YES
1647  007314  032777  002000  171636          BIT     #SW10,@SWR      ;GOTO NEXT TEST?
1648  007322  001407                          BEQ     2$              ;BR IF NO
1649  007324  013737  001360  001126          MOV     NEXT,$LPADR     ;SET FOR NEXT TEST
1650  007332  012706  001120          1$:     MOV     #STACK,SP       ;RESET SP
1651  007336  000177  171564                  JMP     @$LPADR         ;GOTO SPECIFIED TEST
1652  007342  000002                  2$:     RTI                     ;RETURN
1653  007344  000001                  ERTABO: 1
1654  007346     006     002                  .BYTE   6,2
1655  007350  001402                          SAVPC
1656  007352  000001                  XTSTN:  1
1657  007354     002     002                  .BYTE   2,2
1658  007356  001122                          $TSTNM
1659  007360  022737  177570  001160  SERV.G: CMP     #177570,SWR     ;IS THE SWITCH REGISTER HARDWIRED?
1660  007366  001513                          BEQ     6$              ;IF SO, IGNORE ^G
1661  007370  017746  171572                  MOV     @$TKB,-(SP)     ;OTHERWISE, GET THE LAST CHARACTER TYPED
1662  007374  042716  000200                  BIC     #BIT7,(SP)      ;STRIP PARITY(EIGHTH) BIT
1663  007400  122726  000007                  CMPB    #7,(SP)+        ;IS IT ^G?
1664  007404  001104                          BNE     6$              ;IF NOT, IGNORE INPUT
1665  007406  032777  004000  171550          BIT     #4000,@$TKS     ;RX BUSY?
1666  007414  001361                          BNE     SERV.G          ;BR IF YES
1667  007416  017737  171536  007640          MOV     @SWR,90$        ;SAVE (SWR).
1668  007424  013777  007640  171526  1$:     MOV     90$,@SWR        ;
1669  007432  104402  007620                  TYPE    ,89$            ;
1670  007436  104412  007632                  CNVRT   ,88$            ;
1671  007442  104402  007642                  TYPE    ,91$            ;
1672  007446  105777  171512                  TSTB    @$TKS           ;WAIT FOR DONE.
1673  007452  100375                          BPL     .-4             ;
1674  007454  017746  171506                  MOV     @$TKB,-(SP)     ;
1675  007460  042716  000200                  BIC     #BIT7,(SP)      ;
1676  007464  122726  000015                  CMPB    #15,(SP)+       ;
1677  007470  001450                          BEQ     5$              ;
1678  007472  005077  171462                  CLR     @SWR            ;
1679  007476  105777  171466          2$:     TSTB    @$TPS           ;
1680  007502  100375                          BPL     .-4             ;
1681  007504  016677  177776  171460          MOV     -2(SP),@$TPB    ;
1682  007512  000241                          CLC                     ;
1683  007514  006177  171440                  ROL     @SWR            ;
1684  007520  006177  171434                  ROL     @SWR            ;
1685  007524  006177  171430                  ROL     @SWR            ;
1686  007530  103735                          BCS     1$              ;ERROR
1687  007532  026627  177776  000060          CMP     -2(SP),#60      ;
1688  007540  002731                          BLT     1$              ;
1689  007542  026627  177776  000067          CMP     -2(SP),#67      ;
1690  007550  003325                          BGT     1$              ;
1691  007552  042766  177770  177776          BIC     #^C<7>,-2(SP)   ;
1692  007560  056677  177776  171372          BIS     -2(SP),@SWR     ;
1693  007566  105777  171372                  TSTB    @$TKS           ;
1694  007572  100375                          BPL     .-4             ;
1695  007574  017746  171366                  MOV     @$TKB,-(SP)
1696  007600  042716  000200                  BIC     #BIT7,(SP)
1697  007604  122726  000015                  CMPB    #15,(SP)+
```

```
                                        BNE     2$                      ;
                                5$:     TYPE    ,$CRLF                  ;
                                6$:     RTS     PC
                        
                        020200  051450  051127  89$:    .ASCIZ  <200>? (SWR)=/?
                        036451  000057
                                        .EVEN
                                88$:    1
                            006     000         .BYTE   6,0
                        007640                  90$
                        000000          90$:    .WORD   0
                        036457  000057  91$:    .ASCIZ  ?/=/?
                                        .EVEN
                                        .SBTTL  POWER DOWN AND UP ROUTINES
                        
                                        ;;**************************************************************************
                                        ;POWER DOWN ROUTINE
                        012737  010012  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
                        012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
                                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
                                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
                                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
                                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
                        017746  171256          MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
                        010637  010016          MOV     SP,$SAVR6       ;;SAVE SP
                        012737  007720  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
                        000000                  HALT
                        000776                  BR      .-2             ;;HANG UP
                        
                                        ;;**************************************************************************
                                        ;POWER UP ROUTINE
                        012737  010012  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
                        013706  010016          MOV     $SAVR6,SP       ;;GET SP
                        005037  010016          CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
                        005237  010016  1$:     INC     $SAVR6          ;;WAIT FOR THE INC
                        001375                  BNE     1$              ;;OF  WORD
                        012677  171210          MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
                        012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
                        012604                  MOV     (SP)+,R4        ;;POP STACK INTO R4
                        012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
                        012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
                        012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
                        012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
                        012737  007646  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
                        012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
                        104402                  TYPE                    ;;REPORT THE POWER FAILURE
                        010020          $PWRMG: .WORD   MPFAIL          ;;POWER FAIL MESSAGE POINTER
                        012716                  MOV     (PC)+,(SP)      ;;RESTART AT RESTART
                        011434          $PWRAD: .WORD   RESTART         ;;RESTART ADDRESS
                        000002                  RTI
                        000000          $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
                        000776                  BR      .-2             ;;  BEFORE THE POWER DOWN WAS COMPLETE
                        000000          $SAVR6: 0                       ;;PUT THE SP HERE
                        050200  051127  043040  MPFAIL: .ASCIZ  <200>/PWR FAILED. RESTART AT LAST TEST /
```

```
(2)  010063  200  047105  020104  MEPASS: .ASCIZ  <200>/END PASS DZDZA-E  /
(2)  010107  200  052522  047116  MR:     .ASCIZ  <200>/RUNNING    /
(2)  010123  200  051120  043517  MERR2:  .ASCIZ  <200>/PROGRAM INDICATES NO DEVICES PRESENT./
(2)  010172  044600  051516  043125  MERR3:  .ASCIZ  <200>/INSUFFICIENT DATA!/
(2)  010216  046200  041517  020113  MLOCK:  .ASCIZ  <200>/LOCK ON SELECTED TEST/
(2)  010245  103  051123  020072  MCSRX:  .ASCIZ  /CSR: /
(2)  010253  126  041505  020072  MVECX:  .ASCIZ  /VEC: /
(2)  010261  120  051501  042523  MPASSX: .ASCIZ  /PASSES: /
(2)  010272  051105  047522  051522  MERRX:  .ASCIZ  /ERRORS: /
(2)  010303  124  051505  020124  MTSTN:  .ASCIZ  /TEST NO: /
(2)  010315  052  000040  MASTEK: .ASCIZ  /* /
(2)  010320  051600  052105  051440  MNEW:  .ASCIZ  <200>/SET SWITCH REG TO DZ11'S DESIRED ACTIVE./
(2)  010372  041520  020072  000  MERRPC: .ASCIZ  /PC: /
(2)  010377  200  040515  020120  XHEAD:  .ASCIZ  <200>/MAP OF DZ11 STATUS/<200>
(2)  010424  044600  046114  043505  MBADLN: .ASCIZ  <200>/ILLEGAL ENTRY IN STAGGERED MODE/<200>
(2)  .EVEN
(2)  010466  000002  XSTATQ: 2
1754  010470  006  003  .BYTE  6,3
1755  010472  001220  $TMP1
1756  010474  006  002  .BYTE  6,2
1757  010476  001222  $TMP2
1758  .EVEN
1759  ;THIS ROUTINE ESTABLISHES WHICH MAINTENANCE MODE THE DEVICE IS IN
1760  ;-----------------------------------------------------------
1761  ;E=EXTERNAL LOOP BACK
1762  ;I=INTERNAL LOOP BACK
1763  ;S=STAGGERED LOOP BACK
1764  010500  017605  000000  .SETFLG: MOV  @(SP),R5  ;PICK UP ADDRESS OF TAG
1765  010504  042737  000040  010620  BIC  #40,INBUF  ;STRIP LOWER CASE
1766  010512  122737  000105  010620  CMPB  #'E,INBUF  ;IS IT EXTERNAL LOOP BACK ?
1767  010520  001005  BNE  4$  ;NO
1768  010522  013715  010612  MOV  1$,(R5)  ;YES STORE INFO
1769  010526  105037  001417  CLRB  MNTFLG  ;SET MAINT BIT =0
1770  010532  000422  BR  7$  ;GET OUT
1771  010534  122737  000111  010620  4$:  CMPB  #'I,INBUF  ;IS IT INTERNAL LOOP BACK ?
1772  010542  001006  BNE  5$  ;NO
1773  010544  013715  010614  MOV  2$,(R5)  ;YES STORE INFO
1774  010550  112737  000010  001417  MOVB  #MAINT,MNTFLG  ;SET UP THE MAINTENANCE FLAG LOADER
1775  010556  000410  BR  7$  ;GET OUT
1776  010560  122737  000123  010620  5$:  CMPB  #'S,INBUF  ;IS IT STAGGERED LOOP BACK ?
1777  010566  001007  BNE  6$  ;WHAT ?
1778  010570  013715  010616  MOV  3$,(R5)  ;YES STORE INFO
1779  010574  105037  001417  CLRB  MNTFLG  ;ZERO BITS
1780  010600  062716  000002  7$:  ADD  #2,(SP)  ;POP AROUND
1781  010604  000002  RTI
1782  010606  104404  6$:  INSTER  ;RETRY
1783  010610  000733  BR  .SETFLG  ;DITTO
1784  010612  000200  1$:  .WORD  200  ;EXTERNAL = E
1785  010614  000000  2$:  .WORD  0  ;INTERNAL = I
1786  010616  100000  3$:  .WORD  100000  ;STAGGERED = S
1787
1788  ;BUFFERS FOR INPUT-OUTPUT
1789
1790  010620  000000  INBUF:  0
1791  010662  .= +40
1792  010662  000000  TEMP:  0
```

```
 1793            010724                  .=.+40
 1794  010724    000000                  MDATA:  0
 1795            010766                  .=.+40
 1796
 1797  010766    011637  011064          SET.PS: MOV     (SP),3$
 1798  010772    162737  000002  011064          SUB     #2,3$
 1799  011000    017737  000060  011066          MOV     @3$,4$
 1800  011006    022737  106427  011066          CMP     #106427,4$
 1801  011014    001003                          BNE     1$
 1802  011016    011637  011064                  MOV     (SP),3$
 1803  011022    000412                          BR      2$
 1804  011024    022737  106437  011066  1$:     CMP     #106437,4$
 1805  011032    001401                          BEQ     .+4
 1806  011034    000000                          HALT            ;RESERVED INSTRUCTION NOT "MTPS"
 1807  011036    011637  011064                  MOV     (SP),3$
 1808  011042    017737  000016  011064          MOV     @3$,3$
 1809  011050    062716  000002          2$:     ADD     #2,(SP)
 1810  011054    017766  000004  000002          MOV     @3$,2(SP)
 1811  011062    000002                          RTI
 1812  011064    000000                  3$:     0
 1813  011066    000000                  4$:     0
```

```
                                        ;
                                        ;ROUTINE USED TO "CYCLE" THROUGH UP TO SIXTEEN DZ11'S
                                        ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
                                        ;AND RUNS THE SPECIFIED DZ11'S.   THIS ROUTINE *MUST*
                                        ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
                                        ;SETUP NECESSARY.
                                        ;
011070  005737  001404          CYCLE:  TST     DZACTV          ;ARE ANY DZ11'S TO BE TESTED?
011074  001004                          BNE     1$              ;BR IF OK.
011076  104402  010123                  TYPE    ,MERR2          ;NO DZ11'S SELECTED!!
011102  000000                          HALT                    ;STOP THE SHOW.
011104  000776                          BR      .-2             ;DISQUALIFY CONT. SW.
011106  013737  005234  001226  1$:     MOV     $MXCNT,$TIMES   ;RESTORE THE NUMBER OF ITERATIONS TO MAKE
011114  033737  001406  001404          BIT     RUN,DZACTV      ;IS THIS ONE "ACTIVE"
011122  001020                          BNE     2$              ;BR IF GOOD ONE FOUND.
011124  000241                          CLC
011126  006137  001406                  ROL     RUN             ;UPDATE POINTER
011132  005537  001406                  ADC     RUN             ;CATCH CARRY FROM RUN
011136  062737  000014  001412          ADD     #14,ACTIVE      ;UPDATE ADDRESS POINTER.
011144  022737  002000  001412          CMP     #DZ.END,ACTIVE  ;HAVE WE PASSED THE END OF THE MAP?
011152  001355                          BNE     1$              ;IF NO, KEEP GOING; NOT ALL TESTED FOR.
011154  012737  001500  001412          MOV     #DZ.MAP,ACTIVE  ;RESET ADDRESS POINTER.
011162  000751                          BR      1$              ;KEEP LOOKING FOR ACTIVE DZ11
011164  000241                  2$:     CLC
011166  006137  001406                  ROL     RUN             ;UPDATE POINTER.
011172  005537  001406                  ADC     RUN             ;CATCH CARRY.
011176  013700  001412                  MOV     ACTIVE,R0       ;GET ADDRESS POINTER.
011202  062737  000014  001412          ADD     #14,ACTIVE      ;UPDATE.
011210  022737  002000  001412          CMP     #DZ.END,ACTIVE
                                                                ;ALL DONE?
011216  001003                          BNE     3$              ;BR IF NO.
011220  012737  001500  001412          MOV     #DZ.MAP,ACTIVE  ;RESTORE POINTER.
011226  012037  001310          3$:     MOV     (R0)+,$BASE     ;LOAD SYSTEM CTRL. REG
011232  012037  002072                  MOV     (R0)+,DZRIV     ;LOAD VECTOR
011236  012037  027060                  MOV     (R0)+,DZPRT     ;LOAD PRIORITY
011242  113737  027061  001414          MOVB    DZPRT+1,EIAFLG  ;EIA OR 20MA
011250  042737  100000  027060          BIC     #BIT15,DZPRT    ;CLEAR FLAG
011256  012037  001364                  MOV     (R0)+,LINE      ;SET UP LINE DZ LINES ACTIVE
011262  012037  001366                  MOV     (R0)+,PAR       ;SET UP PARAMETERIZATION
011266  012037  001370                  MOV     (R0)+,MODE      ;SET UP MAINTENANCE MODE
011272  004737  026652                  JSR     PC,DZLEV        ;SET UP
011276  005737  000042                  TST     @#42            ;ARE WE UNDER MONITOR CONTROL?
011302  001051                          BNE     4$              ;IF YES, SKIP THIS SETUP
011304  032777  000002  167646          BIT     #SW01,@#SWR     ;IF SW01=1, GET STARTING TEST #
011312  001445                          BEQ     4$              ;BR IF NO TEST IS TO BE INPUTTED
011314  104402  001231          7$:     TYPE    ,$CRLF
011320  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
011322  010303                          MTSTN                   ;POINTER TO MESSAGE TO BE PRINTED
011324  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
011326  000001                          1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
011330  001000                          1000                    ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
011332  001122                          $TSTNM                  ;POINTER TO MAP LOCATION TO BE FILLED
011334      000                         .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
011335      001                         .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
```

```
1870  011336  012700  012374          MOV     #TST1,R0
1871  011342  022710  000004    5$:   CMP     #4,(R0)
1872  011346  001020                  BNE     6$
1873  011350  022760  012737  000002  CMP     #12737,2(R0)
1874  011356  001014                  BNE     6$
1875  011360  023760  001122  000004  CMP     $TSTNM,4(R0)    ;IS THIS THE TEST ^
1876  011366  001010                  BNE     6$              ;IF NOT, DON'T PROCESS NUMBER
1877  011370  010037  001126          MOV     R0,$LPADR       ;SAVE PC
1878  011374  062737  000002  001126  ADD     #2,$LPADR       ;POP OVER SCOPE
1879  011402  104402  001231          TYPE    ,$CRLF
1880  011406  000412                  BR      8$
1881  011410  005720            6$:   TST     (R0)+
1882  011412  020027  023002          CMP     R0,#TLAST+10
1883  011416  001351                  BNE     5$
1884  011420  104402  001230          TYPE    ,$QUES
1885  011424  000733                  BR      7$
1886  011426  012737  012374  001126 4$:    MOV     #TST1,$LPADR    ;PREPARE TEST ADDRESS
1887  011434                    8$:
1888  011434  000177  167466  RESTART:JMP     @$LPADR         ;GO START TESTING.***WARNING!****
1889                                                          ;THIS JUMP IS USED BY POWER UP ROUTINE!!!!
1890
```

```
                                        ;-ROUTINE USED TO SET UP THE DIAGNOSTIC VIA APT.
                                        ; IF BIT7 IN THE ENVIRONMENT MODE ($ENVM) BYTE IS SET,
                                        ; THE PROGRAM WILL LOAD ITS PARAMETERS FROM THE ETABLE.
011440  012700  001500          SETAPT: MOV     #DZ.MAP,R0      ;POINT TO THE DEVICE MAP TABLE
011444  013701  001310                  MOV     $BASE,R1        ;BUILD DEVICE ADDRESSES IN R1
011450  013702  001304                  MOV     $VECT1,R2       ;BUILD DEVICE VECTORS IN R2
011454  042702  177007                  BIC     #^C<770>,R2     ;STRIP AWAY OTHER INFORMATION

011460  113703  001305                  MOVB    $VECT1+1,R3     ;LOAD THE INTERRUPT PRIORITY FROM R3
011464  106003                          RORB    R3              ;ALIGN THE NUMBER
011466  106003                          RORB    R3              ;ALIGN THE NUMBER
011470  106003                          RORB    R3              ;ALIGN THE NUMBER
011472  106003                          RORB    R3              ;ALIGN THE NUMBER
011474  106003                          RORB    R3              ;ALIGN THE NUMBER
011476  042703  177770                  BIC     #^C<7>,R3       ;REMOVE ALL BUT BUS LEVEL NUMBER
011502  012704  001320                  MOV     #$DDW0,R4       ;POINT TO THE BEGINNING OF DEVICE PARAMETERS
011506  013705  001312                  MOV     $DEVM,R5        ;GET THE MAP OF ACTIVE DEVICES
011512  010537  001404                  MOV     R5,DZACTV       ;SAVE THE BIT MAP
011516  006005                  1$:     ROR     R5              ;GET A DEVICE SELECTION BIT
011520  103407                          BCS     3$              ;IF IT IS SELECTED, GO SET UP A MAP
011522  001425                          BEQ     5$              ;IF NO MORE ARE SELECTED, GET OUT OF SETUP
011524  005724                          TST     (R4)+           ;POINT TO NEXT DEVICE DESCRIPTOR
011526  062701  000010          2$:     ADD     #10,R1          ;SET UP THE NEXT ADDRESS
011532  062702  000010                  ADD     #10,R2          ;SET UP THE NEXT VECTOR GROUP
011536  000767                          BR      1$              ;GO SEE IF MORE DEVICES REMAIN
011540  010120                  3$:     MOV     R1,(R0)+        ;LOAD DEVICE ADDRESS
011542  010220                          MOV     R2,(R0)+        ;LOAD THE VECTOR ADDRESS
011544  010320                          MOV     R3,(R0)+        ;LOAD THE INTERRUPT PRIORITY LEVEL
011546  013720  001314                  MOV     $CDW1,(R0)+     ;GET THE NUMBER OF LINES IN OPERATION
011552  012420                          MOV     (R4)+,(R0)+     ;LOAD DEVICE PARAMETERS
011554  100006                          BPL     4$              ;IF 20MA MODE SELECTED, SET IT UP
011556  052760  100000  177772          BIS     #100000,-6(R0)  ;SET THE 20MA FLAG IN DZLVN
011564  042760  100000  177776          BIC     #100000,-2(R0)  ;CLEAR THE FLAG IN DZPARN
011572  005020                  4$:     CLR     (R0)+           ;DEFAULT OPERATION TO INTERNAL MAINTENANCE MODE
011574  000754                          BR      2$              ;GO BUILD THE NEXT ADDRESS
011576  012710  177777          5$:     MOV     #-1,(R0)        ;TERMINATE THE DEVICE MAP
011602  012737  001256  001160          MOV     #$SWREG,SWR     ;SET TO SOFTWARE APT SWITCH REGISTER
011610  000207                          RTS     PC              ;RETURN TO PRINT STATUS TABLE


                                        ;*ROUTINE USED TO "AUTO SIZE" THE DZ11
                                        ;*CSR AND VECTOR.
                                        ;*NOTE:  THE CSR MAY BE ANY WHERE IN THE FLOATING
                                        ;*       ADDRESS RANGE (160000:163700)
                                        ;*       AND THE VECTOR MAY BE ANY WHERE IN THE
                                        ;*       FLOATING VECTOR RANGE (300:770)
                                        ;*

011612                  AUTO.SIZE:
011612  000005                          RESET                   ;INSURE A BUS INIT.
011614  105337  001415                  DECB    INIFLG          ;SHOW THAT I WAS HERE
011620  012702  001500          CSRMAP: MOV     #DZ.MAP,R2      ;LOAD MAP POINTER.
011624  012703  001320                  MOV     #$DDW0,R3       ;POINT TO ETABLE DEVICE DESCRIPTOR WORDS
011630  005022                  1$:     CLR     (R2)+           ;ZERO ENTIRE MAP
011632  022702  002000                  CMP     #DZ.END,R2      ;ALL DONE?
```

```
1947  011636  001374                        BNE     1$              ;BR IF NO
1948  011640  105037  001410                CLRB    DZNUM           ;SET OCTAL NUMBER OF DZ11'S TO 0
1949  011644  012702  001500                MOV     #DZ.MAP,R2
1950  011650  012701  160000                MOV     #160000,R1      ;SET FOR FIRST ADDRESS TO BE TESTED
1951  011654  012737  012174  000004        MOV     #6$,@#4         ;SET FOR NON-EXISTENT DEVICE TIME OUT
1952  011662  052711  000040          2$:   BIS     #BIT5,(R1)      ;TRY TO SET MASTER SCAN ENABLE
1953  011666  052761  000200  000004        BIS     #BIT7,4(R1)     ;TRY TO TRANSMIT ON LINE 7
1954  011674  005000                        CLR     R0              ;USE R0 AS A COUNTER
1955  011676  005711                  7$:   TST     (R1)            ;HAS TRANSMITTER READY COME UP?
1956  011700  100403                        BMI     8$              ;IF SO, GO GET A FINAL CHECK
1957  011702  005300                        DEC     R0              ;REDUCE COUNT. TIME UP?
1958  011704  001374                        BNE     7$              ;IF NOT, KEEP WAITING
1959  011706  000463                        BR      3$              ;ASSUME IT'S NOT A DZ11
1960  011710  032761  000200  000004  8$:   BIT     #BIT7,4(R1)     ;IS LINE 7 ENABLE STILL SET? IT SHOULD BE
1961  011716  001457                        BEQ     3$              ;IF IT'S NOT, ASSUME IT'S NOT A DZ11
1962  011720  032711  000040                BIT     #BIT5,(R1)      ;IS MASTER SCAN ENABLE STILL SET?
1963  011724  001454                        BEQ     3$              ;IF NOT, ASSUME IT'S NOT A DZ11
1964  011726  005000                        CLR     R0
1965  011730  052711  000020                BIS     #20,(R1)        ;SET DEVICE CLEAR
1966  011734  032711  000020                BIT     #20,(R1)        ;SHOULD STAY SET FOR A WHILE IF DZ
1967  011740  001446                        BEQ     3$              ;BR IF NOT DZ11
1968  011742  032711  000020                BIT     #20,(R1)        ;WAIT FOR BIT TO CLEAR
1969  011746  001404                        BEQ     .+12            ;BR WHEN CLEARED
1970  011750  104414                        DELAY
1971  011752  005200                        INC     R0
1972  011754  001372                        BNE     .-12
1973  011756  000437                        BR      3$              ;BIT NOT CLEARED! MUST NOT BE DZ11
1974  011760  005011                        CLR     (R1)            ;GET RID OF MASTER SCAN ENABLE
1975  011762  005061  000004                CLR     4(R1)           ;GET RID OF LINE 7 ENABLE
1976                          ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DZ11 CSR ADDRESS.
1977  011766  010122                        MOV     R1,(R2)+        ;STORE CSR IN CORE TABLE.
1978  011770  005722                        TST     (R2)+           ;POP OVER VECTOR STORE AREA
1979  011772  012722  000005                MOV     #5,(R2)+        ;SET THE DEFAULT BUS LEVEL
1980  011776  052761  177400  000004        BIS     #177400,4(R1)   ;TRY TO SET ALL DTR BITS
1981  012004  032761  177400  000004        BIT     #177400,4(R1)   ;IF ANY SET ASSUME FIA BOARD
1982  012012  001003                        BNE     9$              ;IF NONE SET ASSUME BOARD IS
1983  012014  052762  100000  177776        BIS     #BIT15,-2(R2)   ;20 MA, SET 20 MA FLAG
1984  012022  012722  000377            9$: MOV     #377,(R2)+      ;SET THE DEFAULT LINE SELECTION PARAMETER
1985  012026  012712  017470                MOV     #17470,(R2)     ;SET THE DEFAULT PARAMETERS
1986  012032  012223                        MOV     (R2)+,(R3)+     ;COPY PARAMETERS INTO ETABLE DESCRIPTOR
1987  012034  005022                        CLR     (R2)+           ;SET THE DEFAULT MODE OF OPERATION
1988  012036  012712  177777                MOV     #-1,(R2)        ;TERMINATE LIST
1989  012042  105237  001410                INCB    DZNUM           ;UPDATE DEVICE COUNTER
1990  012046  122737  000020  001410        CMPB    #20,DZNUM       ;ARE MAX. NO. OF DEV FOUND?
1991  012054  001405                        BEQ     100$            ;YES DON'T LOOK FOR ANY MORE.
1992  012056  062701  000010            3$: ADD     #10,R1          ;UPDATE CSR POINTER ADDRESS
1993  012062  022701  163700                CMP     #163700,R1
1994  012066  001275                        BNE     2$              ;BR IF MORE ADDRESS TO CHECK.
1995  012070                          100$:
1996  012070  105737  001410                TSTB    DZNUM           ;WERE ANY DZ11'S FOUND AT ALL?
1997  012074  001432                        BEQ     5$              ;ERROR AUTO SIZER FOUND NO DZ11'S IN THIS SYS.
1998  012076  113701  001410                MOVB    DZNUM,R1
1999  012102  110137  001411                MOVB    R1,SAVNUM       ;SAVE NUMBER OF DEVICES
2000  012106  012737  000001  001404        MOV     #1,DZACTV
2001  012114  005301                  4$:   DEC     R1
2002  012116  001404                        BEQ     98$
```

```
2003  012120  000261                        SEC
2004  012122  006137  001404                ROL     DZACTV
2005  012126  000772                        BR      4$
2006  012130  013737  001500  001310  98$:  MOV     DZCR0,$BASE     ;POINT TO THE ADDRESS OF FIRST DEVICE
2007  012136  013737  001512  001314        MOV     MANTO,$CDWI     ;INDICATE TO ETABLE WHAT MODE IS BEING USED
2008  012144  012737  000006  000004  99$:  MOV     #6,@#4          ;RESTORE TRAP VECTOR
2009  012152  013737  001404  001312        MOV     DZACTV,$DEVM    ;SAVE ACTIVE REGISTER
2010  012160  000410                        BR      VECMAP          ;GO FIND THE VECTOR NOW.
2011  012162  104402  010123          5$:   TYPE    ,MERR2          ;NOTIFY OPR THAT NO DZ11'S FOUND.
2012  012166  005000                        CLR     R0              ;MAKE DATA DISPLAY ZERO
2013  012170  000000                        HALT                    ;STOP THE SHOW
2014  012172  000776                        BR      .-2             ;DISABLE CONT. SW.
2015  012174  012716  012056          6$:   MOV     #3$,(SP)        ;ENTERED BY NON-EXISTENT TIME-OUT
2016  012200  000002                        RTI                     ;RETURN TO MAINSTREAM
2017
2018  012202  012737  000340  000022  VECMAP: MOV   #340,@#22       ;SET IOT TRAP PRIORITY TO 7
2019  012210  012737  012326  000020        MOV     #4$,@#20        ;SET IOT TRAP VECTOR
2020  012216  012702  001500                MOV     #DZ.MAP,R2      ;SET SOFTWARE POINTER
2021  012222  012700  000300                MOV     #300,R0         ;FLOATING VECTORS START HERE.
2022  012226  012701  000302                MOV     #302,R1         ;PC OF IOT INSTR.
2023  012232  010120          1$:   MOV     R1,(R0)+        ;START FILLING VECTOR AREA
2024  012234  012721  000004                MOV     #4,(R1)+        ;WITH .+2; IOT
2025  012240  022021                        CMP     (R0)+,(R1)+     ;ADD 2 TO R0 +R1
2026  012242  020127  001000                CMP     R1,#1000        ;HAS THE VECTOR AREA BEEN EXCEEDED?
2027  012246  101771                        BLOS    1$              ;BR IF MORE TO FILL
2028  012250  013704  001404                MOV     DZACTV,R4       ;STORE TEMPORARILY
2029  012254  000241          2$:   CLC
2030  012256  006004                        ROR     R4              ;BRING OUT A BIT
2031  012260  103036                        BCC     5$              ;BR IF ALL DONE
2032  012262  106427  000000                MTPS    #0              ;ZERO CPU PRIO
2033  012266  012772  040040  000000        MOV     #BIT14+BIT5,@(R2)
2034  012274  011201                        MOV     (R2),R1         ;GET CSR
2035  012276  112761  000200  000004        MOVB    #BIT7,4(R1)     ;SET THE TCR BIT!
2036                                                                ;ATTEMPT TO FORCE AN INTERRUPT
2037  012304  005200                        INC     R0              ;STALL
2038  012306  001376                        BNE     .-2             ;       FOR TIME TO INTERRUPT
2039  012310  012762  000300  000002        MOV     #300,2(R2)      ;NO INTERRUPT ASSUME 300 AND FIX DZ11 LATER
2040  012316  000005          3$:   RESET                           ;INIT
2041  012320  062702  000014                ADD     #14,R2          ;POP SOFTWARE POINTER
2042  012324  000753                        BR      2$              ;KEEP GOING
2043  012326  011662  000002          4$:   MOV     (SP),2(R2)      ;GET VECTOR ADDRESS
2044  012332  162762  000010  000002        SUB     #10,2(R2)       ;POINT BACK TO THE CORRECT VECTOR
2045  012340  042762  000007  000002        BIC     #7,2(R2)        ;CLEAR JUNK
2046  012346  022626                        POP2SP                  ;POP IOT JUNK OFF STACK
2047  012350  012716  012316                MOV     #3$,(SP)        ;SET FOR RETURN
2048  012354  000002                        RTI
2049  012356  013737  001502  001304  5$:   MOV     DZVC0,$VECTI    ;COPY VECTOR OF FIRST DEVICE INTO ETABLE
2050  012364  012737  004772  000020        MOV     #.SCOPE,IOTVEC  ;RESTORE THE SCOPE TRAP
2051  012372  000207                        RTS     PC              ;ALL DONE WITH "AUTO SIZING"
2052
```

```
                                                ;************************ TEST 1 *********************************
                                                ;*THIS TEST PROVES THE SLAVE SYNC RESPONSE
                                                ;*DURING A READ OR WRITE TO THE FOLLOWING ADDRESS:
                                                ;*      DZCSR, DZRBUF, DZTCR, DZMSR
                                        ;:*  TEST 1
                                        ;:*************************************************************
012374  000004                        TST1:   SCOPE
012376  012737  000001  001122                MOV     #1,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
012404  012737  012564  001360                MOV     #TST2,NEXT      ;POINT TO THE STA..T OF THE NEXT TEST
012412  012737  012552  000004                MOV     #5$,4           ;SET TRAP VECTOR
012420  012737  000340  000006                MOV     #PR7,6          ;SET PRIORITY TO LEVEL 7
012426  012737  012434  001362                MOV     #1$,LOCK        ;SET RETURN IF SW09=11
012434  013700  002042                1$:     MOV     DZCSR,R0        ;SET ADDRESS TO TEST
012440  011001                              MOV     (R0),R1         ;READ THE ADDRESS
012442  000240                              NOP                     ;WASTE TIME
012444  005010                              CLR     (R0)            ;WRITE THE ADDRESS
012446  000240                              NOP                     ;WASTE TIME
012450  012737  012456  001362                MOV     #2$,LOCK        ;SET RETURN ADDRESS FOR SW09
012456  013700  002046                2$:     MOV     DZRBUF,R0       ;SET ADDRESS TO TEST
012462  011001                              MOV     (R0),R1         ;READ THE ADDRESS
012464  000240                              NOP
012466  005010                              CLR     (R0)            ;WRITE THE ADDRESS
012470  000240                              NOP                     ;WASTE TIME
012472  012737  012500  001362                MOV     #3$,LOCK        ;SET RETURN ADDRESS FOR SW09
012500  013700  002056                3$:     MOV     DZTCR,R0        ;SET ADDRESS TO TEST
012504  011001                              MOV     (R0),R1         ;READ THE ADDRESS
012506  000240                              NOP
012510  005010                              CLR     (R0)            ;WRITE THE ADDRESS
012512  000240                              NOP
012514  012737  012522  001362                MOV     #4$,LOCK        ;SET RETURN ADDRESS
012522  013700  002062                4$:     MOV     DZMSR,R0        ;SET ADDRESS TO TEST
012526  011001                              MOV     (R0),R1         ;READ FROM ADDRESS
012530  000240                              NOP
012532  005010                              CLR     (R0)            ;WRITE THE ADDRESS
012534  000240                              NOP
012536  012737  000006  000004                MOV     #6,4            ;SET TRAP CATCHER BACK TO NORMAL
012544  005037  000006                      CLR
012550  104400                              ADVANCE                 ;SCOPE THIS TEST
012552  011601                        5$:   MOV     (SP),R1         ;SAVE PC OF TRAP
012554  022626                              CMP     (SP)+,(SP)+     ;POP TRAP OFF STACK
012556  104001                              ERROR   1               ;*NO SLAVE SYNC RESPONSE.
012560  104401                              SCOP1                   ;SW09=1?
012562  000111                              JMP     (R1)            ;RTI
                                                ;************************ TEST 2 *********************************
                                                ;*THIS TEST PROVES THAT BIT "DCLR"
                                                ;*CAN BE SET AND THAT IT WILL CLEAR
                                                ;*BY ITSELF AFTER A PERIOD OF TIME.
                                        ;:*  TEST 2
                                        ;:*************************************************************
012564  000004                        TST2:   SCOPE
012566  012737  000002  001122                MOV     #2,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
012574  012737  012650  001360                MOV     #TST3,NEXT      ;POINT TO THE START OF THE NEXT TEST
012602  013700  002042                        MOV     DZCSR,R0        ;SET POINTER
012606  012705  000020                        MOV     #DCLR,R5        ;SET DCLR
012612  010510                              MOV     R5,(R0)         ;WRITE DCLR INTO DZCSR
```

```
2109  012614  011004                        MOV     (R0),R4         ;READ BACK DZCSR
2110  012616  020504                        CMP     R5,R4           ;DZCSR OK?
2111  012620  001401                        BEQ     1$              ;IF IT IS SET SKIP THE ERROR CALL
2112  012622  104002                        ERROR   2               ;*DCLR SHOULD BE SET..MOMENTARILY
2113                                    ;NOW LETS WATCH IT DISAPPEAR
2114  012624  005002                1$:     CLR     R2              ;SET COUNTER TO 0
2115  012626  005005                        CLR     R5              ;SET EXPECTED TO 0
2116  012630  005003                        CLR     R3              ;DUAL LOOP COUNTER
2117  012632  011004                2$:     MOV     (R0),R4         ;IS DCLR CLEAR?
2118  012634  001405                        BEQ     3$              ;IF YES , GO TO THE NEXT TEST
2119  012636  005203                        INC     R3              ;IF NO,COUNT 1 OF 65535 TICKS
2120                                                                ;THE WORD CREATED BY THE IMMEDIATE 0 WILL BE
2121                                                                ;THE COUNTER
2122  012640  001374                        BNE     2$              ;HAS THE TIME EXPIRED? IF NO, GO TEST BIT AGAIN
2123  012642  005302                        DEC     R2              ;HAS THE TOTAL TIME EXPIRED?
2124  012644  001372                        BNE     2$              ;IF NO, CHECK THE BIT AGAIN
2125  012646  104002                        ERROR   2               ;*DCLR FAILED TO CLEAR
2126  012650                        3$:
2127                                    ;************************ TEST 3 *********************************
2128                                    ;*TEST TO VERIFY THAT BIT "MAINT" CAN
2129                                    ;*BE SET. THEN VERIFY THAT BIT "MAINT" CAN
2130                                    ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
2131                                    ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
2132                                    ;*CLEARED BY A "DEVICE CLEAR"
2133                                ;:*  TEST 3
2134                                ;:*************************************************************
2135  012650  000004                TST3:   SCOPE
2136  012652  012737  000003  001122        MOV     #3,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
2137  012660  012737  012742  001360        MOV     #TST4,NEXT      ;POINT TO THE START OF THE NEXT TEST
2138  012666  013700  002042                MOV     DZCSR,R0        ;GET BASE ADDRESS
2139  012672  012705  000010                MOV     #MAINT,R5       ;SET BIT
2140  012676  010510                        MOV     R5,(R0)         ;SET SET IN DEVICE
2141  012700  011004                        MOV     (R0),R4         ;READ THE BIT FROM DEVICE
2142  012702  020504                        CMP     R5,R4           ;WAS BIT SET?
2143  012704  001401                        BEQ     1$              ;BR IF YES
2144  012706  104002                        ERROR   2               ;*BIT R/W FAILURE
2145  012710  040510                1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
2146  012712  011004                        MOV     (R0),R4         ;READ DEVICE
2147  012714  001404                        BEQ     2$              ;BR IF BITS WERE CLEARED.
2148  012716  010546                        MOV     R5,-(SP)        ;SAVE THE BIT
2149  012720  005005                        CLR     R5              ;SET EXPECTED RESULTS TO 0
2150  012722  104002                        ERROR   2               ;*BIT FAILED TO CLEAR
2151  012724  012605                        MOV     (SP)+,R5        ;RESTORE THE BIT.
2152  012726  010510                2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
2153  012730  104413                        DEVICE.CLR              ;ISSUE DEVICE CLEAR
2154  012732  011004                        MOV     (R0),R4         ;READ THE BIT.
2155  012734  001402                        BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
2156  012736  005005                        CLR     R5              ;SET EXPECTED TO ZERO
2157  012740  104002                        ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
2158  012742                        3$:
2159                                    ;************************ TEST 4 *********************************
2160                                    ;*TEST TO VERIFY THAT BIT "MSENAB" CAN
2161                                    ;*BE SET. THEN VERIFY THAT BIT "MSENAB" CAN
2162                                    ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
2163                                    ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
2164                                    ;*CLEARED BY A "DEVICE CLEAR"
```

```
                                        ;:*  TEST 4
                                        ;:*****************************************************************
                                        TST4:   SCOPE
013742  000004                          
013744  012737  000004  001122          
013752  012737  013034  001360          
013760  013700  002042                  
013764  012705  000040                  
013770  010510                          
013772  011004                          
013774  020504                          
013776  001401                          
013000  104002                          
013002  040510                          1$:     BIC     R5,(R0)
013004  011004                          
013006  001404                          
013010  010546                          
013012  005005                          
013014  104002                          
013016  012605                          
013020  010510                          2$:     MOV     R5,(R0)
013022  104413                          
013024  011004                          
013026  001402                          
013030  005005                          
013032  104002                          
013034                                  3$:
                                        ;************************ TEST 5 *********************************
                                        ;*TEST TO VERIFY THAT BIT "SILOEN" CAN
                                        ;*BE SET. THEN VERIFY THAT BIT "SILOEN" CAN
                                        ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                                        ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                                        ;*CLEARED BY A "DEVICE CLEAR"
                                        ;:*  TEST 5
                                        ;:*****************************************************************
013034  000004                          TST5:   SCOPE
013036  012737  000005  001122          
013044  012737  013126  001360          
013052  013700  002042                  
013056  012705  010000                  
013062  010510                          
013064  011004                          
013066  020504                          
013070  001401                          
013072  104002                          
013074  040510                          1$:     BIC     R5,(R0)
013076  011004                          
013100  001404                          
013102  010546                          
013104  005005                          
013106  104002                          
013110  012605                          
013112  010510                          2$:     MOV     R5,(R0)
013114  104413                          
013116  011004                          
013120  001402                          
013122  005005                          
```

```
                                        MOV     #4,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
                                        MOV     #TST5,NEXT      ;POINT TO THE START OF THE NEXT TEST
                                        MOV     DZCSR,R0        ;GET BASE ADDRESS
                                        MOV     #MSENAB,R5      ;SET BIT
                                        MOV     R5,(R0)         ;SET SET IN DEVICE
                                        MOV     (R0),R4         ;READ THE BIT FROM DEVICE
                                        CMP     R5,R4           ;WAS BIT SET?
                                        BEQ     1$              ;BR IF YES
                                        ERROR   2               ;*BIT R/W FAILURE
                                        BIC     R5,(R0)         ;CLEAR THE BIT.
                                        MOV     (R0),R4         ;READ DEVICE
                                        BEQ     2$              ;BR IF BITS WERE CLEARED.
                                        MOV     R5,-(SP)        ;SAVE THE BIT
                                        CLR     R5              ;SET EXPECTED RESULTS TO 0
                                        ERROR   2               ;*BIT FAILED TO CLEAR
                                        MOV     (SP)+,R5        ;RESTORE THE BIT.
                                        MOV     R5,(R0)         ;SET THE BIT AGAIN
                                        DEVICE.CLR              ;ISSUE DEVICE CLEAR
                                        MOV     (R0),R4         ;READ THE BIT.
                                        BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
                                        CLR     R5              ;SET EXPECTED TO ZERO
                                        ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
                                        MOV     #5,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
                                        MOV     #TST6,NEXT      ;POINT TO THE START OF THE NEXT TEST
                                        MOV     DZCSR,R0        ;GET BASE ADDRESS
                                        MOV     #SILOEN,R5      ;SET BIT
                                        MOV     R5,(R0)         ;SET SET IN DEVICE
                                        MOV     (R0),R4         ;READ THE BIT FROM DEVICE
                                        CMP     R5,R4           ;WAS BIT SET?
                                        BEQ     1$              ;BR IF YES
                                        ERROR   2               ;*BIT R/W FAILURE
                                        BIC     R5,(R0)         ;CLEAR THE BIT.
                                        MOV     (R0),R4         ;READ DEVICE
                                        BEQ     2$              ;BR IF BITS WERE CLE  ED.
                                        MOV     R5,-(SP)        ;SAVE THE BIT
                                        CLR     R5              ;SET EXPECTED RESULTS TO 0
                                        ERROR   2               ;*BIT FAILED TO CLEAR
                                        MOV     (SP)+,R5        ;RESTORE THE BIT.
                                        MOV     R5,(R0)         ;SET THE BIT AGAIN
                                        DEVICE.CLR              ;ISSUE DEVICE CLEAR
                                        MOV     (R0),R4         ;READ THE BIT.
                                        BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
                                        CLR     R5              ;SET EXPECTED TO ZERO
```

```
                                        ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
                                3$:
                                        ;************************ TEST 6 *********************************
                                        ;*TEST TO VERIFY THAT BIT "RIE" CAN
                                        ;*BE SET. THEN VERIFY THAT BIT "RIE" CAN
                                        ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                                        ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                                        ;*CLEARED BY A "DEVICE CLEAR"
                                ;:*  TEST 6
                                ;:*************************************************************
013126  000004                  TST6:   SCOPE
013130  012737  000006  001122          MOV     #6,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
013136  012737  013220  001360          MOV     #TST7,NEXT      ;POINT TO THE START OF THE NEXT TEST
013144  013700  002042                  MOV     DZCSR,R0        ;GET BASE ADDRESS
013150  012705  000100                  MOV     #RIE,R5 ;SET BIT
013154  010510                          MOV     R5,(R0)         ;SET SET IN DEVICE
013156  011004                          MOV     (R0),R4         ;READ THE BIT FROM DEVICE
013160  020504                          CMP     R5,R4           ;WAS BIT SET?
013162  001401                          BEQ     1$              ;BR IF YES
013164  104002                          ERROR   2               ;*BIT R/W FAILURE
013166  040510                  1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
013170  011004                          MOV     (R0),R4         ;READ DEVICE
013172  001404                          BEQ     2$              ;BR IF BITS WERE CLEARED.
013174  010546                          MOV     R5,-(SP)        ;SAVE THE BIT
013176  005005                          CLR     R5              ;SET EXPECTED RESULTS TO 0
013200  104002                          ERROR   2               ;*BIT FAILED TO CLEAR
013202  012605                          MOV     (SP)+,R5        ;RESTORE THE BIT.
013204  010510                  2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
013206  104413                          DEVICE.CLR              ;ISSUE DEVICE CLEAR
013210  011004                          MOV     (R0),R4         ;READ THE BIT.
013212  001402                          BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
013214  005005                          CLR     R5              ;SET EXPECTED TO ZERO
013216  104002                          ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
013220                          3$:
                                        ;************************ TEST 7 *********************************
                                        ;*TEST TO VERIFY THAT BIT "TIE" CAN
                                        ;*BE SET. THEN VERIFY THAT BIT "TIE" CAN
                                        ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                                        ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                                        ;*CLEARED BY A "DEVICE CLEAR"
                                ;:*  TEST 7
                                ;:*************************************************************
013220  000004                  TST7:   SCOPE
013222  012737  000007  001122          MOV     #7,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
013230  012737  013312  001360          MOV     #TST10,NEXT     ;POINT TO THE START OF THE NEXT TEST
013236  013700  002042                  MOV     DZCSR,R0        ;GET BASE ADDRESS
013242  012705  040000                  MOV     #TIE,R5 ;SET BIT
013246  010510                          MOV     R5,(R0)         ;SET SET IN DEVICE
013250  011004                          MOV     (R0),R4         ;READ THE BIT FROM DEVICE
013252  020504                          CMP     R5,R4           ;WAS BIT SET?
013254  001401                          BEQ     1$              ;BR IF YES
013256  104002                          ERROR   2               ;*BIT R/W FAILURE
013260  040510                  1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
013262  011004                          MOV     (R0),R4         ;READ DEVICE
013264  001404                          BEQ     2$              ;BR IF BITS WERE CLEARED.
013266  010546                          MOV     R5,-(SP)        ;SAVE THE BIT
```

```
2277  013270  005005                        CLR     R5              ;SET EXPECTED RESULTS TO 0
2278  013272  104002                        ERROR   2               ;*BIT FAILED TO CLEAR
2279  013274  012605                        MOV     (SP)+,R5        ;RESTORE THE BIT.
2280  013276  010510                2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
2281  013300  104413                        DEVICE.CLR              ;ISSUE DEVICE CLEAR
2282  013302  011004                        MOV     (R0),R4         ;READ THE BIT.
2283  013304  001402                        BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
2284  013306  005005                        CLR     R5              ;SET EXPECTED TO ZERO
2285  013310  104002                        ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
2286  013312                        3$:
2287                                        ;************************ TEST 10 *******************************
2288                                        ;*THIS TESTS THAT ALL OF THE FOLLOWING
2289                                        ;*BITS CAN BE: SET, CLEARED, CLEARED BY "DEVICE CLEAR "
2290                                        ;*BITS TESTED ARE:
2291                                        ;* TCR0, TCR1, TCR2, TCR3, TCR4, TCR5, TCR6, TCR7
2292                                ;;*  TEST 10
2293                                ;;*************************************************************
2294  013312  000004                TST10:  SCOPE
2295  013314  012737  000010  001122        MOV     #10,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
2296  013322  012737  013450  001360        MOV     #TST11,NEXT     ;POINT TO THE START OF THE NEXT TEST
2297  013330  013700  002056                MOV     DZTCR,R0        ;SET DEVICE ADDRESS
2298  013334  012705  000001                MOV     #TCR0,R5        ;SET EXPECTED RESULTS
2299  013340  012737  013346  001362        MOV     #1$,LOCK        ;SET FOR SW09
2300  013346  010510                1$:     MOV     R5,(R0)         ;SET THE BIT
2301  013350  011004                        MOV     (R0),R4         ;READ THE BIT FROM THE DEVICE
2302  013352  042704  177400                BIC     #^C<377>,R4     ;CLEAR HIGH BYTE
2303  013356  020504                        CMP     R5,R4           ;WAS BIT OK?
2304  013360  001401                        BEQ     2$              ;BR IF YES
2305  013362  104002                        ERROR   2               ;*BIT FAILED TO SET.
2306  013364  040510                2$:     BIC     R5,(R0)         ;CLEAR THE BIT
2307  013366  011004                        MOV     (R0),R4         ;READ THE REGISTER
2308  013370  042704  177400                BIC     #^C<377>,R4     ;CLEAR HIGH BYTE
2309  013374  005704                        TST     R4              ;BITS CLEAR?
2310  013376  001404                        BEQ     3$              ;BR IF YES
2311  013400  010546                        MOV     R5,-(SP)        ;SAVE GOOD RESULTS
2312  013402  005005                        CLR     R5              ;SET EXPECTED TO 0
2313  013404  104002                        ERROR   2               ;*REPORT BIT NOT CLEAR
2314  013406  012605                        MOV     (SP)+,R5        ;RESTORE R5
2315  013410  010510                3$:     MOV     R5,(R0)         ;SET THE BIT AGAIN.
2316  013412  104413                        DEVICE.CLR              ;ISSUE DEVICE CLEAR
2317  013414  011004                        MOV     (R0),R4         ;READ THE REGISTER
2318  013416  042704  177400                BIC     #^C<377>,R4     ;CLEAR HIGH BYTE
2319  013422  005704                        TST     R4              ;BITS CLEAR?
2320  013424  001404                        BEQ     4$              ;BR IF YES
2321  013426  010546                        MOV     R5,-(SP)        ;SAVE GOOD RESULTS
2322  013430  005005                        CLR     R5              ;SET EXPECTED TO 0
2323  013432  104002                        ERROR   2               ;*REPORT BIT NOT CLEAR
2324  013434  012605                        MOV     (SP)+,R5        ;RESTORE R5
2325  013436  104401                4$:     SCOP1                   ;LOCK ON BIT? SET SW09=1
2326  013440  106305                        ASLB    R5              ;CHANGE TO NEXT BIT
2327  013442  001341                        BNE     1$              ;CONTINUE TESTING
2328  013444  005037  001362                CLR     LOCK            ;MAKE SURE TIGHT LOOP IS CLEANED UP
2329                                        ;************************ TEST 11 *******************************
2330                                        ;*THIS TESTS THAT ALL OF THE FOLLOWING
2331                                        ;*BITS CAN BE: SET, CLEARED, CLEARED BY "RESET INSTR *NOT* DEVICE CLEAR "
2332                                        ;*BITS TESTED ARE:
```

```
                                        ;* DTR0, DTR1, DTR2, DTR3, DTR4, DTR5, DTR6, DTR7
                                        ;*THIS TEST IS NOT DONE IF MODULE IS 20MA VERSION
                                ;:*  TEST 11
                                ;:*****************************************************************
013450  000004                  TST11:  SCOPE
013452  012737  000011  001122          MOV     #11,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
013460  012737  013632  001360          MOV     #TST12,NEXT     ;POINT TO THE START OF THE NEXT TEST
013466  013700  002056                  MOV     DZTCR,R0        ;SET DEVICE ADDRESS
013472  012705  000400                  MOV     #DTR0,R5        ;SET EXPECTED RESULTS
013476  012737  013514  001362          MOV     #1$,LOCK        ;SET FOR SW09
013504  105737  001414                  TSTB    EIAFLG          ;20MA OR EIA
013510  100001                          BPL     1$              ;BR IF EIA
013512  104400                          ADVANCE                 ;EXIT TEST
013514  010510                  1$:     MOV     R5,(R0)         ;SET THE BIT
013516  011004                          MOV     (R0),R4         ;READ THE BIT FROM THE DEVICE
013520  105004                          CLRB    R4              ;CLEAR LOW BYTE
013522  020504                          CMP     R5,R4           ;WAS BIT OK?
013524  001401                          BEQ     2$              ;BR IF YES
013526  104002                          ERROR   2               ;*BIT FAILED TO SET.
013530  040510                  2$:     BIC     R5,(R0)         ;CLEAR THE BIT
013532  011004                          MOV     (R0),R4         ;READ THE REGISTER
013534  105004                          CLRB    R4              ;CLEAR LOW BYTE
013536  005704                          TST     R4              ;BITS CLEAR?
013540  001404                          BEQ     3$              ;BR IF YES
013542  010546                          MOV     R5,-(SP)        ;SAVE GOOD RESULTS
013544  005005                          CLR     R5              ;SET EXPECTED TO 0
013546  104002                          ERROR   2               ;*REPORT BIT NOT CLEAR
013550  012605                          MOV     (SP)+,R5        ;RESTORE R5
013552  010510                  3$:     MOV     R5,(R0)         ;SET THE BIT AGAIN.
013554  104413                          DEVICE.CLR              ;ISSUE DEVICE CLEAR
013556  011004                          MOV     (R0),R4         ;READ THE REGISTER
013560  105004                          CLRB    R4              ;CLEAR LOW BYTE
013562  030510                          BIT     R5,(R0)         ;WAS BIT CLEARED BY DEVICE.CLR?
013564  001001                          BNE     4$              ;BR IF NO (IT SHOULDN'T BE CLEAR)
013566  104002                          ERROR   2               ;*BIT CLEARED BY DEVICE.CLR
013570  104401                  4$:     SCOP1                   ;LOCK ON BIT? SW09=1
013572  006305                          ASL     R5              ;CHANGE TO NEXT BIT
013574  001347                          BNE     1$              ;IF NOT DONE LOOP
013576  012710  177400                  MOV     #177400,(R0)    ;SET ALL DTR BITS
013602  005005                          CLR     R5              ;CLEAR LOCATION FOR ERROR PRINTOLT
013604  005227  000000          5$:     INC     #0              ;ACT DELAY LOOP FOR
013610  001375                          BNE     5$              ;RESET INSTRUCTION
013612  000005                          RESET                   ;ISSUE A BUS INIT
013614  011004                          MOV     (R0),R4         ;READ REGISTER
013616  105004                          CLRB    R4              ;CLEAR LOW BYTE
013620  005704                          TST     R4              ;DTR BITS CLEAR?
013622  001401                          BEQ     .+4             ;IF YES CONTINUE
013624  104002                          ERROR   2               ;IF NO PRINT ERROR
013626  005037  001362                  CLR     LOCK            ;MAKE SURE TIGHT LOOP IS CLEANED UP
                                ;************************ TEST 12 *********************************
                                ;*THIS TEST PERFORMS RESET TESTING &
                                ;*TESTING OF WRITE ONLY OR READ ONLY BIT
                                ;* TEST BITS "RDONE, BIT11, BIT10, BIT9, BIT8, BIT2, BIT1
                                ;*             BIT0, SILOAL" ARE READ ONLY AND THAT TRDY IS
                                ;*             ZERO'UNTIL A LINE IS SELECTED AND MSENAB IS SET.
                                ;*
```

```
                                        ;:*  TEST 12
                                        ;:*****************************************************************
                                        ;TST12:  SCOPE
013632  000004                                   MOV     #12,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
013634  012737  000012  001122                   MOV     #TST13,NEXT     ;POINT TO THE START OF THE NEXT TEST
013642  012737  013750  001360                   MOV     DZCSR,R0        ;SET ADDRESS TO R0
013650  013700  002042                           CLR     R5              ;SET EXPECTED TO 0
013654  005005                                   MOV     #RDONE+BIT11+BIT10+BIT9+BIT8+BIT2+BIT1+BIT0+SILOAL,(R0)
013656  012710  027607                                                   ;WRITE THE BITS
013662  011004                                   MOV     (R0),R4         ;READ BACK THE BITS
013664  001401                                   BEQ     1$              ;BR IF NONE ARE SET.
013666  104002                                   ERROR   2               ;*BITS WERE SET.
013670  012710  100000                   1$:     MOV     #TRDY,(R0)      ;ATTEMPT TO WRITE TRDY
013674  011004                                   MOV     (R0),R4         ;READ TRDY
013676  001401                                   BEQ     2$              ;BR IF NOT SET
013700  104002                                   ERROR   2               ;*
013702  012705  100000                   2$:     MOV     #TRDY,R5        ;SET EXPECTED BIT
013706  005077  166140                           CLR     @DZLPR          ;LOAD LINE 0
013712  052777  000001  166136                   BIS     #TCR0,@DZTCR    ;SET TCR BIT
013720  052710  000040                           BIS     #MSENAB,(R0)
013724  052705  000040                           BIS     #MSENAB,R5      ;SET SCAN ENABLE
013730  005002                                   CLR     R2              ;SET COUNTER TO ZERO
013732  011004                           3$:     MOV     (R0),R4         ;READ THE REGISTER
013734  020504                                   CMP     R5,R4           ;BIT SET?
013736  001404                                   BEQ     4$              ;BR IF YES
013740  104414                                   DELAY                   ;STALL TIME
013742  005202                                   INC     R2              ;UPDATE COUNTER
013744  001372                                   BNE     3$              ;BR IF COUNTER NOT DONE.
013746  104002                                   ERROR   2               ;*TRDY NOT SET!
013750                                   4$:
                                                 ;************************ TEST 13 *********************************
                                                 ;*THIS TEST PERFORMS RESET TESTING AND
                                                 ;*TESTING OF READ ONLY AND WRITE ONLY BITS
                                                 ;* IN REGISTER DZCSR
                                                 ;*VERIFY THAT "TIE", "SILOEN", "RIE", "MSENAB", "MAINT"
                                                 ;*ARE THE ONLY R/W BITS IN THE DZCSR.
                                                 ;*THEN VERIFY THAT A RESET WILL CLEAR THESE BITS
                                                 ;*THIS TEST ALSO CHECKS BYTE OPERATIONS ON THE CSR
                                        ;:*  TEST 13
                                        ;:*****************************************************************
                                        ;TST13:  SCOPE
013750  000004                                   MOV     #13,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
013752  012737  000013  001122                   MOV     #TST14,NEXT     ;POINT TO THE START OF THE NEXT TEST
013760  012737  014100  001360                   DEVICE.CLR
013766  104413                                   MOV     DZCSR,R0        ;SET UP FOR ERROR MESSAGE
013770  013700  002042                           MOV     #^C<DCLR>,(R0)  ;TRY TO WRITE
013774  012710  177757                           MOV     #TIE!SILOEN!RIE!MSENAB!MAINT,R5 ;MAKE EXPECTED
014000  012705  050150                           MOV     (R0),R4         ;ACTUAL
014004  011004                                   CMP     R4,R5           ;CMP EXPECTED VS ACTUAL
014006  020405                                   BEQ     1$              ;YES
014010  001401                                   ERROR   2               ;*NO
014012  104002                           1$:     CLRB    (R0)            ;CLEAR LOWER BYTE OF CSR
014014  105010                                   CLRB    R5              ;SET EXPECTED
014016  105005                                   MOV     (R0),R4         ;READ CSR BITS
014020  011004                                   CMP     R4,R5           ;COMPARE ACTUAL TO EXPECTED
014022  020405                                   BEQ     3$              ;BRANCH IF SAME
014024  001401
```

```
                                        ERROR   2               ;OTHERWISE PRINT ERROR
                                3$:     MOV     #↑C<DCLR>,(R0)  ;RESET CSR BITS
                                        CLRB    @#DZCSR         ;CLEAR HIGH BYTE OF CSR
                                        MOV     #RIE!MSENAB!MAINT,R5
                                                                ;SET R5 TO EXPECTED RESULTS
                                        MOV     (P0),R4         ;READ CSR
                                        CMP     R4,R5           ;ACTUAL = EXPECTED?
                                        BEQ     4$              ;BRANCH IF SAME
                                        ERROR   2               ;OTHERWISE PRINTOUT ERROR
                                4$:     MOV     #↑C<DCLR>,(R0)  ;RESET CSR BITS
                                        CLR     R5              ;SET R5 TO EXPECTED RESULTS
                                5$:     INC     #0              ;DELAY TIMER FOR
                                        BNE     5$              ;ACT-11 COMPATIBILITY
                                        RESET                   ;ISSUE BUS INIT
                                        MOV     (R0),R4         ;READ CSR REGISTER
                                        BEQ     2$              ;BRANCH IF CSR IS CLEAR
                                        ERROR   2               ;IF NOT PRINT ERROR
                                2$:
                                        ;************************ TEST 14 *********************************
                                        ;*THIS TEST PERFORMS RESET TESTING AND
                                        ;*TESTING OF READ ONLY REGISTER DZRBUF
                                        ;*AND TESTING OF WRITE ONLY REGISTER DZLPR
                                ;;*  TEST 14
                                ;;*************************************************************
                                ↑ST14:  SCOPE
                                        MOV     #14,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
                                        MOV     #↑ST15,NEXT     ;POINT TO THE START OF THE NEXT TEST
                                        DEVICE.CLR              ;CLEAR DZ11
                                        MOV     DZRBUF,R0               ;SET UP FOR ERROR MESSAGE
                                        MOV     (R0),R5         ;SET EXPECTED
                                        MOV     #-1,@#DZLPR     ;TRY TO WRITE ALL 1'S
                                        MOV     (R0),R4         ;ACTUAL
                                        BIC     #DVALID!BIT11,R5 ;DITTO
                                        CMP     R4,R5           ;CMP ACTUAL VS EXPECTED
                                        BEQ     1$              ;IF YES,GO CONTINUE PROCESSING
                                        ERROR   2               ;*ERROR- BIT PATTERN NOT CORRECT
                                1$:     MOV     R4,R3           ;GET A COPY OF THE ACTUAL BIT PATTERN
                                        COM     R3              ;GET THE LOGICAL INVERSE OF THE BIT PATTERN
                                        MOV     R3,@#DZLPR      ;TRY TO WRITE
                                        MOV     (R0),R4         ;ACTUAL
                                        CMP     R4,R5           ;CMP ACTUAL VS EXPECTED
                                        BEQ     2$              ;IF YES, GET OUT OF THIS TEST
                                        ERROR   2               ;*NO
                                2$:
                                        ;************************ TEST 15 *********************************
                                        ;*THIS TEST PERFORMS RESET TESTING AND
                                        ;*TESTING OF READ ONLY REGISTER DZMSR
                                        ;*AND TESTING OF WRITE ONLY REGISTER DZTDR
                                ;;*  TEST 15
                                ;;*************************************************************
                                ↑ST15:  SCOPE
                                        MOV     #15,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
                                        MOV     #↑ST16,NEXT     ;POINT TO THE START OF THE NEXT TEST
                                        DEVICE.CLR              ;CLEAR DZ11
                                        MOV     DZMSR,R0                ;SET UP FOR ERROR MESSAGE
                                        MOV     (R0),R5         ;SET EXPECTED
```

```
2501  014216  012777  177777  165642          MOV     #-1,@DZTDR      ;TRY TO WRITE ALL 1'S
2502  014224  011004                          MOV     (R0),R4         ;ACTUAL
2503  014226  020405                          CMP     R4,R5           ;CMP ACTUAL VS EXPECTED
2504  014230  001401                          BEQ     1$              ;IF YES,GO CONTINUE PROCESSING
2505  014232  104002                          ERROR   2               ;*ERROR- BIT PATTERN NOT CORRECT
2506  014234  010403                  1$:     MOV     R4,R3           ;GET A COPY OF THE ACTUAL BIT PATTERN
2507  014236  005103                          COM     R3              ;GET THE LOGICAL INVERSE OF THE BIT PATTERN
2508  014240  010377  165622                  MOV     R3,@DZTDR       ;TRY TO WRITE
2509  014244  011004                          MOV     (R0),R4         ;ACTUAL
2510  014246  020405                          CMP     R4,R5           ;CMP ACTUAL VS EXPECTED
2511  014250  001401                          BEQ     2$              ;IF YES, GET OUT OF THIS TEST
2512  014252  104002                          ERROR   2               ;*NO
2513  014254                          2$:
2514
2515
2516                                          ;************************ TEST 16 *******************************
2517                                          ;*VERIFY THAT IF WE ARE IN "STAGGERED" MODE
2518                                          ;*THAT SETTING "DTR" FOR A LINE WILL
2519                                          ;*BRING UP "RING" AND "CARRIER" FOR THE
2520                                          ;*ASSOCIATED LINE IN WHICH WE ARE STAGGERED!
2521                                          ;* LINE0 DTR= LINE1 RING AND CARRIER
2522                                          ;* LINE1 DTR= LINE0 RING AND CARRIER
2523                                          ;* LINE2 DTR= LINE3 RING AND CARRIER
2524                                          ;* LINE3 DTR= LINE 4 RING AND CARRIER
2525                                          ;*                E*C...
2526
2527                                  ;:*  TEST 16
2528                                  ;:*************************************************************************
2529  014254  000004                  TST16:  SCOPE
2530  014256  012737  000016  001122          MOV     #16,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
2531  014264  012737  014450  001360          MOV     #TST17,NEXT     ;POINT TO THE START OF THE NEXT TEST
2532  014272  012737  014344  001362          MOV     #1$,LOCK        ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
2533  014300  105737  001414                  TSTB    EIAFLG          ;EIA OR 20MA?
2534  014304  100001                          BPL     10$             ;BR IF EIA
2535  014306  104400                          ADVANCE                 ;EXIT TEST
2536  014310  013700  002062          10$:    MOV     DZMSR,R0        ;SET REGISTER
2537  014314  104413                          DEVICE.CLR              ;INIT DZ11
2538  014316  005003                          CLR     R3              ;ZERO LINE NUMBER
2539  014320  012702  000001                  MOV     #1,R2           ;SET POINTER
2540  014324  005737  001370                  TST     MODE            ;ARE WE IN STAGGERED MODE?
2541  014330  100405                          BMI     1$              ;YES WE ARE!
2542  014332  013737  001360  001126          MOV     NEXT,$LPADR     ;LEAVE THIS TEST! NOT STAGGERED
2543  014340  000177  164562                  JMP     @$LPADR         ;EXIT
2544  014344  130237  001364          1$:     BITB    R2,LINE         ;TEST THIS LINE?
2545  014350  001004                          BNE     3$              ;YES
2546  014352  005203                  2$:     INC     R3              ;LINE #
2547  014354  106302                          ASLB    R2              ;GET NEXT LINE
2548  014356  103372                          BCC     1$              ;KEEP TESTING
2549  014360  104400                          ADVANCE                 ;ADVANCE THIS TEST
2550  014362  010204                  3$:     MOV     R2,R4           ;SAVE BINARY BIT FOR LINE #
2551  014364  032703  000001                  BIT     #BIT0,R3        ;GET STAGGERED COMPANION LINE
2552  014370  001402                          BEQ     4$              ;BR IF LINE EVEN
2553  014372  006204                          ASR     R4              ;ADJUST LINE
2554  014374  000401                          BR      5$              ;
2555  014376  006304                  4$:     ASL     R4              ;ADJUST LINE
2556  014400  005005                  5$:     CLR     R5              ;SET EXPECTED
```

```
2557  014402  150405                          BISB    R4,R5           ;
2558  014404  000305                          SWAB    R5              ;
2559  014406  150405                          BISB    R4,R5           ;
2560  014410  150277  165444                  BISB    R2,@#HDZTCR     ;SET DTR
2561  014414  104414                          DELAY                   ;CABLE DELAY
2562  014416  011004                          MOV     (R0),R4         ;READ MSR REGISTER
2563  014420  020504                          CMP     R5,R4           ;OK?
2564  014422  001401                          BEQ     6$              ;YES
2565  014424  104002                          ERROR   2               ;*ERROR IN RING OR CARRIER
2566  014426  140277  165426          6$:     BICB    R2,@#HDZTCR     ;CLEAR DTR
2567  014432  104414                          DELAY                   ;CABLE DELAY
2568  014434  011004                          MOV     (R0),R4         ;READ MSR
2569  014436  001402                          BEQ     7$              ;BR IF THEY CLEARED
2570  014440  005005                          CLR     R5              ;SET EXPECTED TO 0
2571  014442  104002                          ERROR   2               ;*BITS NOT CLEARED
2572  014444  104401                  7$:     SCOPE                   ;LOCK ON SIGNAL?
2573  014446  000741                          BR      2$              ;CONTINUE TEST
2574
2575                                          ;********************** TEST 17 ******************************
2576                                          ;*TEST TO VERIFY THAT IF IN "EXTERNAL"
2577                                          ;*MODE; SETTING DTR FOR SELECTED LINES
2578                                          ;*WILL BRING UP "CARRIER" AND "RING"
2579                                          ;*FOR THAT SAME LINE. NOTE: IF YOU HAVE
2580                                          ;*SELECTED MODE AS "EXTERNAL", THE H325 TEST CONNECTER
2581                                          ;*MUST BE USED ON ALL SPECIFIED LINES.
2582                                          ;*LINES MAY BE SPECIFIED BY SWR03=1
2583                                          ;*AND SWR00=1 AT START TIME OR ALTERING
2584                                          ;*STATUS MAP.
2585                                  ;;*  TEST 17
2586                                  ;;*************************************************************
2587  014450  000004                  TST17:  SCOPE
2588  014452  012737  000017  001122          MOV     #17,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
2589  014460  012737  014606  001360          MOV     #TST20,NEXT     ;POINT TO THE START OF THE NEXT TEST
2590  014466  012737  014522  001362          MOV     #3$,LOCK        ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
2591  014474  105737  001370                  TSTB    MODE            ;EXTERNAL?
2592  014500  100401                          BMI     2$              ;BR IF YES
2593  014502  104400                  1$:     ADVANCE                 ;EXIT TEST
2594  014504  105737  001414          2$:     TSTB    EIAFLG          ;YOU BETTER BE IN
2595  014510  100774                          BMI     1$              ;EIA MODE FOR THIS TEST.
2596  014512  013700  002062                  MOV     DZMSR,R0        ;SET REGISTER
2597  014516  012702  000001                  MOV     #1,R2           ;SET LINE POINTER
2598  014522  130237  001364          3$:     BITB    R2,LINE         ;LINE SELECTED?
2599  014526  001003                          BNE     5$              ;BR IF YES
2600  014530  106302                  4$:     ASLB    R2              ;NEXT LINE
2601  014532  103373                          BCC     3$              ;CONTINUE TEST
2602  014534  104400                          ADVANCE                 ;ADVANCE THIS TEST
2603  014536  005005                  5$:     CLR     R5              ;SET EXPECTED
2604  014540  150205                          BISB    R2,R5           ;
2605  014542  000305                          SWAB    R5              ;
2606  014544  150205                          BISB    R2,R5           ;
2607  014546  150277  165306                  BISB    R2,@#HDZTCR     ;SET DTR
2608  014552  104414                          DELAY                   ;CABLE DELAY
2609  014554  011004                          MOV     (R0),R4         ;READ MSR
2610  014556  020504                          CMP     R5,R4           ;BITS OK?
2611  014560  001401                          BEQ     6$              ;BR IF YES
2612  014562  104002                          ERROR   2               ;CARRIER OR RING ERROR
```

```
2613  014564  140277  165270          6$:     BICB    R2,@#DZTCR      ;CLEAR DTR
2614  014570  104414                          DELAY                   ;CABLE DELAY
2615  014572  011004                          MOV     (R0),R4         ;READ MSR
2616  014574  001402                          BEQ     7$              ;BR IF BITS CLEARED
2617  014576  005005                          CLR     R5              ;CLEAR EXPECTED LOC.
2618  014600  104002                          ERROR   2               ;BITS NOT CLEARED.
2619  014602  104401                  7$:     SCOPE                   ;LOCK ON LINE?
2620  014604  000751                          BR      4$              ;CONTINUE TEST
2621
2622                                          ;************************ TEST 20 *******************************
2623                                          ;* THIS TEST VERIFIES THAT TRDY IS SET WHEN A LINE
2624                                          ;* IS READY TO BE LOADED, AND THAT THE LINE SPECI-
2625                                          ;* FIED IN BITS 8-10 OF DZCSR CORRESPOND
2626                                          ;* TO THE LINE SELECTED IN DZTCR
2627                                  ;;*  TEST 20
2628                                  ;;*************************************************************
2629  014606  000004                  TST20:  SCOPE
2630  014610  012737  000020  001122          MOV     #20,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
2631  014616  012737  014732  001360          MOV     #TST21,NEXT     ;POINT TO THE START OF THE NEXT TEST
2632  014624  104413                          DEVICE.CLR              ;ISSUE A "DEVICE CLEAR" (RESET)
2633  014626  013700  002042                  MOV     DZCSR,R0        ;SET POINTER
2634  014632  012705  100040                  MOV     #MSENAB!TRDY,R5 ;START THE EXPECTED LINE NUMBER AT 0
2635  014636  005037  001372                  CLR     SAVLIN          ;SET UP FOR ERROR PRINTOUTS
2636  014642  012702  000001                  MOV     #1,R2           ;USING R2 AS A BIT POINTER, POINT TO LINE 0
2637  014646  130237  001364          1$:     BITB    R2,LINE         ;IS THIS LINE SELECTED?
2638  014652  001420                          BEQ     5$              ;IF NO, SKIP THE STARTUP
2639  014654  050277  165176          2$:     BIS     R2,@DZTCR       ;SET THE GO BIT FOR THIS LINE
2640  014660  052710  000040                  BIS     #MSENAB,(R0)    ;START THE SCANNER
2641  014664  005004                          CLR     R4              ;SET FOR DELAY
2642  014666  032710  100000          3$:     BIT     #TRDY,(R0)      ;TX READY?
2643  014672  001004                          BNE     4$              ;BR IF YES
2644  014674  104414                          DELAY                   ;DELAY
2645  014676  005204                          INC     R4              ;COUNTER
2646  014700  001372                          BNE     3$              ;BR IF <>0!
2647  014702  104003                          ERROR   3               ;*TX NOT READY!
2648  014704  011004                  4$:     MOV     (R0),R4         ;GET THE LINE POINTED TO BY THE SCANNER
2649  014706  020405                          CMP     R4,R5           ;IS THE LINE NUMBER WHAT IT SHOULD BE?
2650  014710  001401                          BEQ     5$              ;IF YES,GO WORK ON THE NEXT LINE
2651  014712  104002                          ERROR   2               ;*LINE NUMBER DID NOT MATCH TCR BIT
2652  014714  062705  000400          5$:     ADD     #400,R5         ;POINT TO THE NEXT EXPECTED LINE
2653  014720  104413                          DEVICE.CLR              ;ISSUE A "DEVICE CLEAR" (RESET)
2654  014722  005237  001372                  INC     SAVLIN          ;ADJUST FOR NEXT LINE
2655  014726  106302                          ASLB    R2              ;POINT TO THE NEXT LINE.ARE ALL LINES TESTED?
2656  014730  103346                          BCC     1$              ;IF NOT, GO DO THE NEXT LINE
2657  014732                          6$:
2658                                          ;************************ TEST 21 *******************************
2659                                          ;*TEST TO TRANSMIT ONE CHAR AND
2660                                          ;*RECEIVE ONE CHAR ON ONE LINE
2661                                          ;*AT A TIME. THE CHAR IS "252" AND
2662                                          ;*ALL SELECTED LINES WILL BE TURNED ON
2663                                          ;*ONE AT A TIME. THIS IS THE FIRST TIME ANY
2664                                          ;*DATA IS CHECKED IN THE RECEIVER.
2665                                          ;*USING SWITCH NINE WITH THIS TEST CREATES A TIGHT SCOPE LOOP
2666                                          ;*WHICH TRANSMITS A STEADY STREAM OF CHARACTERS.
2667                                  ;;*  TEST 21
2668                                  ;;*************************************************************
```

```
2669  014732  000004                    TST21:  SCOPE
2670  014734  012737  000021  001122            MOV     #21,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
2671  014742  012737  015270  001360            MOV     #TST22,NEXT     ;POINT TO THE START OF THE NEXT TEST
2672  014750  012737  015246  001362            MOV     #16$,LOCK       ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
2673  014756  104417                            DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
2674  014760  013701  001366                    MOV     PAR,R1          ;PICK UP PARAMETERS
2675  014764  012702  000001                    MOV     #1,R2           ;PICK UP INIT POINTER
2676  014770  030237  001364            1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
2677  014774  001402                            BEQ     2$              ;NO
2678  014776  010177  165050                    MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
2679  015002  005201                    2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
2680  015004  106302                            ASLB    R2              ;GOT 'EM ALL ?
2681  015006  103370                            BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
2682  015010  005037  001372                    CLR     SAVLIN          ;CLEAR LINE # INDICATOR
2683  015014  012702  000001                    MOV     #1,R2           ;LINE POINTER
2684  015020  052777  000040  165014            BIS     #MSENAB,@DZCSR  ;START SCANNER
2685  015026  030237  001364            3$:     BIT     R2,LINE         ;VALID LINE ?
2686  015032  001462                            BEQ     14$             ;NO SET UP NEXT LINE
2687  015034  010277  165016                    MOV     R2,@DZTCR       ;SET TCR BIT
2688  015040  032777  000200  164774    4$:     BIT     #RDONE,@DZCSR   ;IS REC DONE = 0 ?
2689  015046  001401                            BEQ     5$              ;IF YES, ALLOW TIME FOR TRDY TO SET
2690  015050  104020                            ERROR   20              ;*REC DONE SHOULD = 0
2691  015052  005005                    5$:     CLR     R5
2692  015054  032777  100000  164760    6$:     BIT     #TRDY,@DZCSR
2693  015062  001004                            BNE     7$
2694  015064  104414                            DELAY
2695  015066  105205                            INCB    R5
2696  015070  001371                            BNE     6$
2697  015072  104003                            ERROR   3               ;*TRDY FAILED TO SET!
2698  015074  112777  000252  164764    7$:     MOVB    #252,@DZTDR     ;LOAD CHARACTER
2699  015102  013705  001372                    MOV     SAVLIN,R5       ;MAKE EXPECTED LINE #
2700  015106  105737  001371                    TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MODE?
2701  015112  001406                            BEQ     10$             ;IF NOT, SKIP STAGGERED SETUP
2702
2703                                            ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
2704
2705  015114  006205                            ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
2706  015116  103402                            BCS     8$              ;IF IT IS SET, GO CLEAR IT
2707  015120  000261                            SEC                     ;IF IT IS CLEAR SET IT HERE
2708  015122  000401                            BR      9$              ;SKIP THE CLEARING
2709  015124  000241                    8$:     CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
2710  015126  006105                    9$:     ROL     R5              ;GET THE NEW BIT BACK INTO R5
2711  015130  000305                    10$:    SWAB    R5              ;MOVE THE LINE NUMBER TO THE UPPER BYTE
2712  015132  152705  000252                    BISB    #252,R5         ;ADD CHARACTER
2713  015136  052705  100000                    BIS     #DVALID,R5      ;ADD DATA VALID
2714  015142  005003                            CLR     R3
2715  015144  032777  000200  164670    11$:    BIT     #RDONE,@DZCSR
2716  015152  001004                            BNE     12$
2717  015154  104414                            DELAY
2718  015156  005203                            INC     R3
2719  015160  001371                            BNE     11$
2720  015162  104004                            ERROR   4               ;*RDONE FAILED TO SET!
2721  015164  017704  164656            12$:    MOV     @DZRBUF,R4      ;LOAD THE VALUE ACTUALLY RECEIVED
2722  015170  020405                            CMP     R4,R5           ;COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
2723  015172  001401                            BEQ     13$             ;IF YES, GO DO THE NEXT LINE
2724  015174  104006                            ERROR   6               ;*NO DATA/CONTENTS DID NOT COMPARE
```

```
2725  015176  104401                    13$:    SCOP1                   ;CHECK TO SEE IF SWITCH NINE IS SET
2726  015200  040277  164652            14$:    BIC     R2,@DZTCR       ;CLEAR TCR BIT FOR THAT LINE.
2727  015204  005237  001372            15$:    INC     SAVLIN          ;INC EXPECTED LINE
2728  015210  013700  001372                    MOV     SAVLIN,R0       ;SET UP CHARACTER OFFSET
2729  015214  006300                            ASL     R0              ;MAKE THE OFFSET A POWER OF TWO
2730  015216  106302                            ASLB    R2              ;SHIFT THE LINE POINTER. ARE WE ALL DONE?
2731  015220  103302                            BCC     3$              ;IF NO, GO AROUND AGAIN FOR NEXT LINE
2732  015222  005003                            CLR     R3              ;THIS CODE HAS BEEN INSERTED
2733  015224  104414                    17$:    DELAY                   ;TO DETECT A PROBLEM FOUND IN FAULT
2734  015226  105203                            INCB    R3              ;INSERTION. IF AN ERROR OCCURS MORE
2735  015230  001375                            BNE     17$             ;THAN ONE WORD WAS RECIEVED ON
2736  015232  032777  000200  164602          BIT     #RDONE,@DZCSR   ;LINE 7.
2737  015240  001401                            BEQ     18$
2738  015242  104020                            ERROR   20
2739  015244  104400                    18$:    ADVANCE                 ;GO TO NEXT TEST
2740
2741                                    ;TIGHT SCOPE LOOP FOR THIS TEST. LOOP TRANSMITS CHARACTERS ONLY
2742
2743  015246  032777  100000  164566    16$:    BIT     #TRDY,@DZCSR    ;IS TRANSMITTER READY?
2744  015254  001774                            BEQ     16$             ;IF NOT, WAIT FOR IT
2745  015256  112777  000252  164602            MOVB    #252,@DZTDR     ;LOAD THE CHARACTER
2746  015264  104401                            SCOP1                   ;LOOP AGIN IF SW09=1
2747  015266  000744                            BR      14$             ;OTHERWISE, GO PICK UP THE TEST NORMALLY
2748
2749                                            ;************************ TEST 22 *********************************
2750                                            ;* THIS TEST PROVES THAT THE TRANSMITTER TRANSMITS
2751                                            ;*CHARACTERS (FLAG MODE)AND THE RECEIVER RECEIVES (FLAG MODE)
2752                                            ;*(ONE LINE AT A TIME   BASED UPON VALID LINES)
2753                                            ;*THIS IS THE FIRST TIME THAT ALL DATA IS CHECKED
2754                                    ;:*  TEST 22
2755                                    ;:*************************************************************
2756  015270  000004                    TST22:  SCOPE
2757  015272  012737  000022  001122            MOV     #22,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
2758  015300  012737  015616  001360            MOV     #TST23,NEXT     ;POINT TO THE START OF THE NEXT TEST
2759  015306  012737  015422  001362            MOV     #4$,LOCK        ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
2760  015314  104417                            DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
2761  015316  013701  001366                    MOV     PAR,R1          ;PICK UP PARAMETERS
2762  015322  012702  000001                    MOV     #1,R2           ;PICK UP INIT POINTER
2763  015326  030237  001364            1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
2764  015332  001402                            BEQ     2$              ;NO
2765  015334  010177  164512                    MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
2766  015340  005201                    2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
2767  015342  106302                            ASLB    R2              ;GOT 'EM ALL ?
2768  015344  103370                            BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
2769  015346  005037  001372                    CLR     SAVLIN          ;CLEAR LINE # INDICATOR
2770  015352  012700  001422                    MOV     #TDO,R0         ;POINT TO THE DATA AREA
2771  015356  005020                            CLR     (R0)+           ;CLEAR A DATA WORD
2772  015360  022700  001462                    CMP     #STOP,R0        ;FINISHED ?
2773  015364  001374                            BNE     .-6             ;NO
2774  015366  005000                            CLR     R0              ;CLEAR OFFSET
2775  015370  013737  002046  001400            MOV     DZRBUF,REGIST   ;SAVE FOR ERROR MSG
2776  015376  012702  000001                    MOV     #1,R2           ;LINE POINTER
2777  015402  052777  000040  164432            BIS     #MSENAB,@DZCSR  ;START SCANNER
2778  015410  030237  001364            3$:     BIT     R2,LINE         ;VALID LINE ?
2779  015414  001465                            BEQ     14$             ;NO SET UP NEXT LINE
2780  015416  010277  164434                    MOV     R2,@DZTCR       ;SET TCR BIT
```

```
2781  015422  032777  000200  164412  4$:     BIT     #RDONE,@DZCSR   ;IS REC DONE = 0 ?
2782  015430  001401                          BEQ     5$              ;IF YES, ALLOW TIME FOR TRDY TO SET
2783  015432  104020                          ERROR   20              ;*REC DONE SHOULD = 0
2784  015434  005005                  5$:     CLR     R5
2785  015436  032777  100000  164376  6$:     BIT     #TRDY,@DZCSR
2786  015444  001004                          BNE     7$
2787  015446  104414                          DELAY
2788  015450  105205                          INCB    R5
2789  015452  001371                          BNE     6$
2790  015454  104003                          ERROR   3               ;*TRDY FAILED TO SET!
2791  015456  116077  001422  164402  7$:     MOVB    TDO(R0),@DZTDR  ;LOAD CHARACTER
2792  015464  013705  001372                  MOV     SAVLIN,R5       ;MAKE EXPECTED LINE #
2793  015470  105737  001371                  TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MODE?
2794  015474  001406                          BEQ     10$             ;IF NOT, SKIP STAGGERED SETUP
2795
2796                                          ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
2797
2798  015476  006205                          ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
2799  015500  103402                          BCS     8$              ;IF IT IS SET, GO CLEAR IT
2800  015502  000261                          SEC                     ;IF IT IS CLEAR SET IT HERE
2801  015504  000401                          BR      9$              ;SKIP THE CLEARING
2802  015506  000241                  8$:     CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
2803  015510  006105                  9$:     ROL     R5              ;GET THE NEW BIT BACK INTO R5
2804  015512  000305                  10$:    SWAB    R5              ;MOVE THE LINE NUMBER TO THE UPPER BYTE
2805  015514  156005  001422                  BISB    TDO(R0),R5      ;ADD CHARACTER
2806  015520  052705  100000                  BIS     #DVALID,R5      ;ADD DATA VALID
2807  015524  005003                          CLR     R3
2808  015526  032777  000200  164306  11$:    BIT     #RDONE,@DZCSR
2809  015534  001004                          BNE     12$
2810  015536  104414                          DELAY
2811  015540  005203                          INC     R3
2812  015542  001371                          BNE     11$
2813  015544  104004                          ERROR   4               ;*RDONE FAILED TO SET!
2814  015546  017704  164274          12$:    MOV     @DZRBUF,R4      ;LOAD THE VALUE ACTUALLY RECEIVED
2815  015552  020405                          CMP     R4,R5           ;COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
2816  015554  001401                          BEQ     13$             ;IF YES, GO DO THE NEXT LINE
2817  015556  104006                          ERROR   6               ;*NO DATA/CONTENTS DID NOT COMPARE
2818  015560  104401                  13$:    SCOP1                   ;CHECK TO SEE IF SWITCH NINE IS SET
2819  015562  105260  001422                  INCB    TDO(R0)         ;INCREMENT BINARY PATTERN FOR THIS LINE
2820  015566  001315                          BNE     4$              ;GO 'ROUND AGAIN FOR NEXT CHARACTER
2821  015570  040277  164262          14$:    BIC     R2,@DZTCR       ;CLEAR TCR BIT FOR THAT LINE.
2822  015574  005237  001372          15$:    INC     SAVLIN          ;INC EXPECTED LINE
2823  015600  013700  001372                  MOV     SAVLIN,R0       ;SET UP CHARACTER OFFSET
2824  015604  006300                          ASL     R0              ;MAKE THE OFFSET A POWER OF TWO
2825  015606  106302                          ASLB    R2              ;SHIFT THE LINE POINTER. ARE WE ALL DONE?
2826  015610  103277                          BCC     3$              ;IF NO, GO AROUND AGAIN FOR NEXT LINE
2827  015612  005037  001362                  CLR     LOCK            ;MAKE SURE LOCK IS CLEAR FOR NEXT TEST
2828
2829
2830                                          ;************************ TEST 23 *********************************
2831                                          ;*THIS TEST WILL PROVE THAT EACH RECEIVING LINE CAN
2832                                          ;*BE DISABLED BY SETTING THE RCVON BIT TO ZERO
2833                                          ;*FOR EACH LINE IN THE LPR REGISTER.  IT ALSO
2834                                          ;*VERIFIES THAT MASTER CLEAR WILL ZERO DVALID FOR
2835                                          ;*CHARACTERS STORED IN THE SILO.
2836                                  ;:*  TEST 23
```

```
                                ;;*****************************************************************
015616 000004                   ↑ST23:  SCOPE
015620 012737 000023 001122             MOV     #23,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
015626 012737 016150 001360             MOV     #↑ST24,NEXT     ;POINT TO THE START OF THE NEXT TEST
015634 105037 001420                    CLRB    DONFLG          ;INITIALIZE FOR FIRST TEST LOOP
015640 005037 001372                    CLR     SAVLIN          ;ZERO LINE NO. FOR ERROR REPORT
015644 104417                           DC_ASM                  ;EXECUTE MASTER CLEAR
015646 013701 001366                    MOV     PAR,R1          ;STORE DEFAULT PARAMETERS
015652 042701 010000                    BIC     #RCVON,R1       ;CLEAR RCVON BIT
015656 012702 000001            1$:     MOV     #1,R2           ;INIT LINE POINTER
015662 010177 164164            2$:     MOV     R1,@DZLPR       ;LOAD LINE PARAMETER REGISTER
015666 005201                           INC     R1              ;SET R1 FOR NEXT LINE
015670 106302                           ASLB    R2              ;SHIFT R2 TO NEXT LINE
015672 103373                           BCC     2$              ;ALL LINES LOADED?
015674 012701 000252                    MOV     #252,R1         ;LOAD TRANSMITTING CHARACTER
015700 013702 001364                    MOV     LINE,R2         ;COPY ACTIVE LINE BITS
015704 010277 164146                    MOV     R2,@DZTCR       ;LOAD TCR BITS
015710 052777 000040 164124             BIS     #MSENAB,@DZCSR  ;SET SCANNER
015716 005005                   3$:     CLR     R5              ;INIT DELAY COUNTER
015720 005777 164116            4$:     TST     @DZCSR          ;TRDY SET?
015724 100404                           BMI     5$              ;IF YES BRANCH
015726 104414                           DELAY                   ;IF NOT THEN WAIT
015730 005205                           INC     R5              ;INCREMENT DELAY COUNTER
015732 001372                           BNE     4$              ;DELAY DONE?
015734 104003                           ERROR   3               ;IF YES TRDY FAILED TO SET
015736 117705 164102            5$:     MOVB    @HDZCSR,R5      ;MOVE LINE NO. INTO R5
015742 012703 000001                    MOV     #1,R3           ;INIT TCR POINTER
015746 042705 177770                    BIC     #↑C<7>,R5       ;ISOLATE LINE NO.
015752 001403                           BEQ     21$             ;IF LINE 0 GO TEST TRANSM. FLAG
015754 106303                   20$:    ASLB    R3              ;POINT R3 TO NEXT TCR BIT
015756 005305                           DEC     R5              ;DECREMENT R5 UNTIL R3 POINTS
015760 001375                           BNE     20$             ;TO CORRECT TCR BIT
015762 030302                   21$:    BIT     R3,R2           ;HAS THIS LINE BEEN SERVICED?
015764 001007                           BNE     6$              ;IF NOT GO SEND CHARACTER
015766 140377 164064                    BICB    R3,@DZTCR       ;IF YES CLEAR TCR BIT
015772 001351                           BNE     3$              ;IF MORE LINES SET BRANCH
015774 105737 001420                    TSTB    DONFLG          ;IF ALL LOADED IS THIS SECOND PASS
016000 001040                           BNE     12$             ;IF YES BRANCH TO SECOND PART OF TEST
016002 000404                           BR      7$              ;OTHERWISE CONTINUE WITH FIRST PART
016004 110177 164056            6$:     MOVB    R1,@DZTDR       ;TRANSMIT CHARACTER
016010 040302                           BIC     R3,R2           ;CLEAR FLAG FOR THIS LINE
016012 000741                           BR      3$              ;GO WAIT FOR NEXT LINE
016014 005077 164036            7$:     CLR     @DZTCR          ;CLEAR TCR BITS
016020 005005                           CLR     R5              ;CLEAR DELAY COUNTER
016022 104414                   8$:     DELAY                   ;WAIT FOR LAST CHARACTER
016024 005205                           INC     R5              ;INCREMENT DELAY COUNTER
016026 001375                           BNE     8$              ;IF NOT FINISHED CONTINUE WAITING
016030 105777 164006                    TSTB    @DZCSR          ;RDONE BIT SET?
016034 100003                           BPL     10$             ;IF NO CONTINUE
016036 005037 001372                    CLR     SAVLIN          ;IF YES SET LINE NO. TO ZERO
016042 104020                           ERROR   20              ;AND PRINT ERROR
016044 017704 163776            10$:    MOV     @DZRBUF,R4      ;READ SILO
016050 100007                           BPL     11$             ;IF DVALID IS ZERO BRANCH
016052 000304                           SWAB    R4              ;IF SET THEN
016054 042704 177770                    BIC     #↑C<7>,R4       ;ISOLATE LINE NO. IN R4
016060 010437 001372                    MOV     R4,SAVLIN       ;SET SAVLIN FOR ERROR REPORT
```

```
2893  016064  104017                          ERROR   17              ;DATA VALID SHOULD NOT BE SET
2894  016066  000766                          BR      10$             ;GO READ SILO AGAIN
2895  016070  105237  001420          11$:    INCB    DONFLG          ;PREPARE FOR SECOND PART OF TEST
2896  016074  013701  001366                  MOV     PAR,R1          ;MOVE DEFAULT PARAMETERS TO R1
2897  016100  000666                          BR      1$              ;GO LOAD LPR REGISTER
2898  016102  005005                  12$:    CLR     R5              ;INIT DELAY COUNTER
2899  016104  104414                  13$:    DELAY                   ;WAIT FOR LAST CHARACTER
2900  016106  005205                          INC     R5              ;TO BE RECEIVED
2901  016110  001375                          BNE     13$             ;DELAY FINISHED?
2902  016112  104413                          DEVICE.CLR              ;IF YES EXECUTE MASTER CLEAR
2903  016114  000240                          NOP
2904  016116  000240                          NOP
2905  016120  105777  163716                  TSTB    @DZCSR          ;RDONE SET?
2906  016124  100003                          BPL     14$             ;IF NOT BRANCH
2907  016126  005037  001372                  CLR     SAVLIN          ;IF YES THEN PRINT OUT
2908  016132  104020                          ERROR   20              ;REPORT
2909  016134  017704  163706          14$:    MOV     @DZRBUF,R4      ;READ SILO
2910  016140  100003                          BPL     15$             ;DATA VALID SET?
2911  016142  005037  001372                  CLR     SAVLIN          ;IF YES THEN PRINT OUT
2912  016146  104017                          ERROR   17              ;ERROR REPORT
2913  016150                          15$:
2914
2915                                          ;************************ TEST 24 *******************************
2916                                          ;*THIS TEST WILL PROVE THAT:
2917                                          ;* 1) THE TRANSMITTER "BREAK BIT" WORKS
2918                                          ;* 2) THE RECEIVER CAN FLAG "FRAMING ERRORS"
2919                                          ;* 3) THE RECEIVER CAN FLAG "PARITY ERRORS"
2920                                          ;*ONLY ONE LINE AT A TIME WILL BE EXERCISED.
2921                                          ;*THIS TEST WILL NOT BE  EXERCISED UNLESS
2922                                          ;*CONNECTED BY AN H325, H3271, OR H3190 CONNECTOR
2923                                  ;:*  TEST 24
2924                                  ;:*************************************************************
2925  016150  000004                  TST24:  SCOPE
2926  016152  012737  000024  001122          MOV     #24,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
2927  016160  012737  016426  001360          MOV     #TST25,NEXT     ;POINT TO THE START OF THE NEXT TEST
2928  016166  012737  016264  001362          MOV     #3$,LOCK        ;SET FOR LOOP
2929  016174  005737  001370                  TST     MODE            ;ARE WE RUNNING IN INTERNAL MODE?
2930  016200  001510                          BEQ     12$             ;IF SO, SKIP THIS TEST
2931  016202  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
2932  016204  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
2933  016210  052701  000300                  BIS     #ODDPAR!PARITY,R1 ;FORCE ODD PARITY
2934  016214  012700  000001                  MOV     #1,R0           ;PICK UP INIT POINTER
2935  016220  030037  001364          1$:     BIT     R0,LINE         ;SHOULD THIS LINE BE SET UP ?
2936  016224  001402                          BEQ     2$              ;IF NOT,DON'T SET IT UP
2937  016226  010177  163620                  MOV     R1,@DZLPR       ;OTHERWISE, SET UP LINE PARAMETERS
2938  016232  005201                  2$:     INC     R1
2939  016234  106300                          ASLB    R0              ;GOT 'EM ALL ?
2940  016236  103370                          BCC     1$              ;NO
2941  016240  005037  001372                  CLR     SAVLIN          ;CLEAR LINE #
2942  016244  012702  000001                  MOV     #1,R2           ;LINE POINTER
2943  016250  052777  000040  163564          BIS     #MSENAB,@DZCSR  ;SET MASTER SCAN ENABLE
2944  016256  013737  002046  001400          MOV     DZRBUF,REGIST   ;SAVE FOR ERRR MESSAGE
2945  016264  030237  001364          3$:     BIT     R2,LINE
2946  016270  001446                          BEQ     11$
2947  016272  010277  163560                  MOV     R2,@DZTCR       ;SET TCR BIT
2948  016276  110277  163566                  MOVB    R2,@HDZTDR      ;SET BREAK BIT
```

```
2949  016302  112777  000377  163556  4$:     MOVB    #377,@DZTDR     ;LOAD CHARACTER
2950  016310  013705  001372                  MOV     SAVLIN,R5       ;MAKE EXPECTED DATA
2951  016314  105737  001371                  TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MODE?
2952  016320  001406                          BEQ     7$              ;IF NOT, SKIP STAGGERED SETUP
2953
2954                                          ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
2955
2956  016322  006205                          ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
2957  016324  103402                          BCS     5$              ;IF IT IS SET, GO CLEAR IT
2958  016326  000261                          SEC                     ;IF IT IS CLEAR SET IT HERE
2959  016330  000401                          BR      6$              ;SKIP THE CLEARING
2960  016332  000241                  5$:     CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
2961  016334  006105                  6$:     ROL     R5              ;GET THE NEW BIT BACK INTO R5
2962  016336  000305                  7$:     SWAB    R5              ;PUT LINE NUMBER IN UPPER BYTE
2963  016340  052705  130000                  BIS     #DVALID!PARER!FRMERR,R5 ;ADD EXPECTED
2964  016344  005004                          CLR     R4
2965  016346  032777  000200  163466  8$:     BIT     #RDONE,@DZCSR
2966  016354  001004                          BNE     9$
2967  016356  104414                          DELAY
2968  016360  005204                          INC     R4
2969  016362  001371                          BNE     8$
2970  016364  104004                          ERROR   4               ;*RDONE FAILED TO SET!
2971  016366  017704  163454          9$:     MOV     @DZRBUF,R4      ;ACTUAL
2972  016372  020405                          CMP     R4,R5           ;CMP ACTUAL VS EXPECTED. DO THEY MATCH?
2973  016374  001401                          BEQ     10$             ;IF YES, GO CLEAN UP
2974  016376  104006                          ERROR   6               ;*DATA/CONTENTS FAILED TO COMPARE
2975  016400  105077  163464          10$:    CLRB    @HDZTDR         ;CLEAR BREAK BITS
2976  016404  104401                          SCOP1                   ;LOOP?
2977  016406  005237  001372          11$:    INC     SAVLIN          ;INC LINE #
2978  016412  040277  163440                  BIC     R2,@DZTCR       ;CLEAR TCR BIT
2979  016416  106302                          ASLB    R2
2980  016420  103321                          BCC     3$
2981  016422  005037  001362          12$:    CLR     LOCK            ;MAKE SURE LOCK IS CLEAR FOR NEXT TEST
2982                                          ;************************ TEST 25 ******************************
2983                                          ;* THIS TEST VERIFIES THAT THE DEVICE DOES NOT INTERRUPT
2984                                          ;*WHILE THE PROCESSOR STATUS IS SET EXACTLY
2985                                          ;*TO WHAT THE DZ11 PRIORITY IS SET TO.
2986                                          ;*DEFAULT PRIORITY IS AT  5 (240).
2987                                  ;:*  TEST 25
2988                                  ;:*************************************************************
2989  016426  000004                  TST25:  SCOPE
2990  016430  012737  000025  001122          MOV     #25,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
2991  016436  012737  016736  001360          MOV     #TST26,NEXT     ;POINT TO THE START OF THE NEXT TEST
2992  016444  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
2993  016446  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
2994  016452  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
2995  016456  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
2996  016462  001402                          BEQ     2$              ;NO
2997  016464  010177  163362                  MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
2998  016470  005201                  2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
2999  016472  106302                          ASLB    R2              ;GOT 'EM ALL ?
3000  016474  103370                          BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
3001  016476  005037  001372                  CLR     SAVLIN          ;CLEAR LINE # INDICATOR
3002  016502  106437  027060                  MTPS    @#DZPRT         ;SET CPU STATUS TO DZ11 PRIO,
3003  016506  113777  001364  163342          MOVB    LINE,@DZTCR     ;ENABLE THE VALID LINES
3004  016514                  3$:
```

```
 3005  016514  012777  016604  163354          MOV     #6$,@DZTIV      ;SET UP THE TRANSMITTER INTERRUPT VECTOR
 3006  016522  012777  016612  163342          MOV     #7$,@DZRIV      ;SET UP THE RECEIVER INTERRUPT VECTOR
 3007  016530  013777  027060  163336          MOV     DZPRT,@DZRIS    ;SET THE INTERRUPT VECTOR STATUS
 3008  016536  013777  027060  163334          MOV     DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
 3009  016544  052777  040040  163270          BIS     #TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
 3010  016552  005005                          CLR     R5
 3011  016554  032777  100000  163260  4$:     BIT     #TRDY,@DZCSR
 3012  016562  001403                          BEQ     5$
 3013  016564  000240                          NOP
 3014  016566  000240                          NOP
 3015  016570  000412                          BR      8$
 3016  016572  104414                  5$:     DELAY
 3017  016574  005205                          INC     R5
 3018  016576  001366                          BNE     4$
 3019  016600  104003                          ERROR   3               ;*TRDY NOT SET!
 3020  016602  000405                          BR      8$
 3021  016604  104010                  6$:     ERROR   10              ;*TRANSMITTER SHOULD NOT INTERRUPT
 3022  016606  022626                          CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
 3023  016610  000402                          BR      8$              ;CONTINUE TEST
 3024  016612  104012                  7$:     ERROR   12              ;*RECEIVER SHOULD NOT INTERRUPT
 3025  016614  022626                          CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
 3026  016616  042777  040000  163216  8$:     BIC     #TIE,@DZCSR     ;RESET TRANSMITTER INTERRUPT ENABLE
 3027  016624  012777  016722  163244          MOV     #11$,@DZTIV     ;SET UP THE TRANSMITTER INTERRUPT VECTOR
 3028  016632  012777  016730  163232          MOV     #12$,@DZRIV     ;SET UP THE RECEIVER INTERRUPT VECTOR
 3029  016640  013777  027060  163226          MOV     DZPRT,@DZRIS    ;SET THE INTERRUPT VECTOR STATUS
 3030  016646  013777  027060  163224          MOV     DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
 3031  016654  052777  000140  163160          BIS     #RIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
 3032  016662  113777  001422  163176          MOVB    TDO,@DZTDR      ;PUT ANY RANDOM CHARACTER IN TRANSMITTER BUFFER
 3033  016670  005005                          CLR     R5
 3034  016672  032777  000200  163142  9$:     BIT     #RDONE,@DZCSR
 3035  016700  001403                          BEQ     10$
 3036  016702  000240                          NOP
 3037  016704  000240                          NOP
 3038  016706  000412                          BR      13$
 3039  016710  104414                  10$:    DELAY
 3040  016712  005205                          INC     R5
 3041  016714  001366                          BNE     9$
 3042  016716  104004                          ERROR   4               ;*NO RX DONE! (NOT SET)
 3043  016720  000405                          BR      13$             ;CONTINUE TEST
 3044  016722  104010                  11$:    ERROR   10              ;*TRANSMITTER SHOULD NOT INTERRUPT
 3045  016724  022626                          CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
 3046  016726  000402                          BR      13$             ;CONT TEST
 3047  016730  104012                  12$:    ERROR   12              ;*RECEIVER SHOULD NOT INTERRUPT
 3048  016732  022626                          CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
 3049  016734                          13$:
 3050  016734  104413                          DEVICE.CLR              ;ISSUE DEVICE CLEAR (RESET)
 3051                                          ;************************ TEST 26 ******************************
 3052                                          ;* THIS TEST VERIFIES THAT THE DEVICE DOES INTERRUPT
 3053                                          ;*WHILE THE PROCESSOR STATUS IS SET TO EXACTLY
 3054                                          ;*ONE LEVEL LOWER THAN THE DZ11. DZ11 PRIORITY
 3055                                          ;*DEFAULT TO LEVEL 5 MINUS ONE LEVEL IS LEVEL 4.
 3056                                  ;:*  TEST 26
 3057                                  ;:**************************************************************
 3058  016736  000004                  TST26:  SCOPE
 3059  016740  012737  000026  001122          MOV     #26,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
 3060  016746  012737  017264  001360          MOV     #TST27,NEXT     ;POINT TO THE START OF THE NEXT TEST
```

```
3061  016754  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3062  016756  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
3063  016762  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
3064  016766  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
3065  016772  001402                          BEQ     2$              ;NO
3066  016774  010177  163052                  MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
3067  017000  005201                  2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
3068  017002  106302                          ASLB    R2              ;GOT 'EM ALL ?
3069  017004  103370                          BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
3070  017006  005037  001372                  CLR     SAVLIN          ;CLEAR LINE # INDICATOR
3071  017012  106437  027062                  MTPS    @#LESS1         ;MAKE CPU ONE LEVEL LOWER THAN DZ11
3072  017016  113777  001364  163032         MOVB    LINE,@DZTCR     ;ENABLE THE VALID LINES
3073  017024                          3$:
3074  017024  012777  017116  163044         MOV     #6$,@DZTIV      ;SET UP THE TRANSMITTER INTERRUPT VECTOR
3075  017032  012777  017134  163032         MOV     #7$,@DZRIV      ;SET UP THE RECEIVER INTERRUPT VECTOR
3076  017040  013777  027060  163026         MOV     DZPRT,@DZRIS    ;SET THE INTERRUPT VECTOR STATUS
3077  017046  013777  027060  163024         MOV     DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
3078  017054  052777  040040  162760         BIS     #TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
3079  017062  005005                          CLR     R5
3080  017064  032777  100000  162750  4$:     BIT     #TRDY,@DZCSR
3081  017072  001404                          BEQ     5$
3082  017074  000240                          NOP
3083  017076  000240                          NOP
3084  017100  104007                          ERROR   7               ;* RANSMITTER FAILED TO INTERRUPT
3085  017102  000416                          BR      8$
3086  017104  104414                  5$:     DELAY
3087  017106  005205                          INC     R5
3088  017110  001365                          BNE     4$
3089  017112  104003                          ERROR   3               ;*TRDY NOT SET!
3090  017114  000411                          BR      8$
3091  017116  022626                  6$:     POP2SP                  ;REMOVE THE INTERRUPT FROM THE STACK
3092  017120  042777  040000  162714         BIC     #TIE,@DZCSR     ;DON'T LET ANY MORE INTERRUPTS OCCUR
3093  017126  106437  027062                  MTPS    @#LESS1         ;MAKE CPU ONE LEVEL LOWER THAN DZ11
3094  017132  000402                          BR      8$              ;RETURN TO THE NORMAL FLOW
3095  017134  104012                  7$:     ERROR   12              ;*RECEIVER SHOULD NOT INTERRUPT
3096  017136  022626                          CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
3097  017140  042777  040000  162674  8$:     BIC     #TIE,@DZCSR     ;RESET TRANSMITTER INTERRUPT ENABLE
3098  017146  012777  017246  162722         MOV     #11$,@DZTIV     ;SET UP THE TRANSMITTER INTERRUPT VECTOR
3099  017154  012777  017254  162710         MOV     #12$,@DZRIV     ;SET UP THE RECEIVER INTERRUPT VECTOR
3100  017162  013777  027060  162704         MOV     DZPRT,@DZRIS    ;SET THE INTERRUPT VECTOR STATUS
3101  017170  013777  027060  162702         MOV     DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
3102  017176  052777  000140  162636         BIS     #RIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
3103  017204  113777  001422  162654         MOVB    TDO,@DZTDR      ;PUT ANY RANDOM CHARACTER IN TRANSMITTER BUFFER
3104  017212  005005                          CLR     R5
3105  017214  032777  000200  162620  9$:     BIT     #RDONE,@DZCSR
3106  017222  001404                          BEQ     10$
3107  017224  000240                          NOP
3108  017226  000240                          NOP
3109  017230  104011                          ERROR   11              ;*RECEIVER FAILED TO INTERRUPT
3110  017232  000413                          BR      13$
3111  017234  104414                  10$:    DELAY
3112  017236  005205                          INC     R5
3113  017240  001365                          BNE     9$
3114  017242  104004                          ERROR   4               ;*NO RX DONE! (NOT SET)
3115  017244  000406                          BR      13$             ;CONTINUE TEST
3116  017246  104010                  11$:    ERROR   10              ;*TRANSMITTER SHOULD NOT INTERRUPT
```

```
                                        CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
                                        BR      13$             ;CONT TEST
                                12$:    POP2SP                  ;REMOVE THE INTERRUPT FROM THE STACK
                                        CLR     @DZCSR          ;DON'T ALLOW ANY MORE INTERRUPTS
                                13$:
                                        DEVICE.CLR              ;ISSUE DEVICE CLEAR (RESET)

                                        ;************************ TEST 27 ********************************
                                        ;*THIS TEST VERIFIES THAT THE RECEIVER WILL
                                        ;*INTERRUPT BEFORE THE TRANSMITTER EVEN
                                        ;*THOUGH THE TRANSMITTER WAS ENABLED
                                        ;*FIRST.  SET PS TO LEVEL 7;
                                        ;*GET RDONE AND TRDY TO SET;
                                        ;*SET TX IE AND RX IE;
                                        ;*CLEAR PS AND EXPECT RX TO INTERRUPT FIRST
                                ;:*  TEST 27
                                ;;*************************************************************
017264  000004                  TST27:  SCOPE
017266  012737  000027  001122          MOV     #27,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
017274  012737  017716  001360          MOV     #TST30,NEXT     ;POINT TO THE START OF THE NEXT TEST
017302  104417                        DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
017304  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
017310  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
017314  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
017320  001402                          BEQ     2$              ;NO
017322  010177  162524                  MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
017326  005201                  2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
017330  106302                          ASLB    R2              ;GOT 'EM ALL ?
017332  103370                          BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
017334  005037  001372                  CLR     SAVLIN          ;CLEAR LINE # INDICATOR
017340  012777  017570  162524          MOV     #8$,@DZRIV      ;SETUP INTERRUPT STUFF
017346  013777  027060  162520          MOV     DZPRT,@DZRIS    ;
017354  012777  017660  162514          MOV     #12$,@DZTIV     ;
017362  013777  027060  162510          MOV     DZPRT,@DZTIS    ;
017370  052777  000040  162444          BIS     #MSENAB,@DZCSR
017376  012702  000001                  MOV     #1,R2           ;LINE POINTER
017402  030237  001364          3$:     BIT     R2,LINE         ;VALID LINE ?
017406  001004                          BNE     4$
017410  005237  001372                  INC     SAVLIN
017414  106302                          ASLB    R2
017416  000771                          BR      3$
017420  106427  000340          4$:     MTPS    #PR7
017424  000240                          NOP
017426  000240                          NOP
017430  110277  162422                  MOVB    R2,@DZTCR       ;SET TCR BIT
017434  005777  162406                  TST     @DZRBUF         ;VALID DATA?
017440  100001                          BPL     .+4             ;IT BETTER NOT BE SET
017442  104017                          ERROR   17              ;DATA VALID SHOULD NOT BE SET
017444  105777  162372          5$:     TSTB    @DZCSR          ;RECEIVER DONE ?
017450  100001                          BPL     .+4
017452  104020                          ERROR   20              ;RECEIVER DONE BIT SHOULD NOT BE SET
017454  005005                          CLR     R5
017456  005004                          CLR     R4
017460  005777  162356          99$:    TST     @DZCSR          ;WAIT FOR TRDY
017464  100404                          BMI     100$            ;BR IF READY
017466  104414                          DELAY                   ;STALL TIME
```

```
3173  017470  005204                          INC     R4              ;
3174  017472  001372                          BNE     99$
3175  017474  104003                          ERROR   3               ;TRDY FAILED TO SET
3176  017476  105077  162364          100$:   CLRB    @DZTDR
3177  017502  005004                          CLR     R4
3178  017504  032777  000200  162330  6$:     BIT     #RDONE,@DZCSR
3179  017512  001004                          BNE     7$
3180  017514  104414                          DELAY
3181  017516  005204                          INC     R4
3182  017520  001371                          BNE     6$
3183  017522  104004                          ERROR   4               ;*RDONE FAILED TO SET!
3184  017524  005777  162312          7$:     TST     @DZCSR          ;TRANS DONE BIT = 1 ?
3185  017530  100401                          BMI     .+4             ;YES
3186  017532  104003                          ERROR   3               ;*NO   TRANS DONE FAILED TO SET
3187                                          ;NOW THAT BOTH TRANSMITTER AND RECEIVER DONE BIT =1
3188                                          ;SET INTERRUPT ENABLES AND WATCH THE FUR FLY
3189  017534  052777  040000  162300          BIS     #TIE,@DZCSR
3190  017542  052777  000100  162272          BIS     #RIE,@DZCSR
3191  017550  106427  000000                  MTPS    #0
3192  017554  000240                          NOP
3193  017556  000240                          NOP
3194  017560  104007                          ERROR   7               ;*TRANSMITTER FAILED TO INTERRUPT
3195  017562  104011                          ERROR   11              ;*RECEIVER FAILED TO INTERRUPT
3196                                          ;CHECK BR LEVEL
3197  017564  000137  017664                  JMP     13$     ;GET OUT
3198
3199                                          ;RECEIVER INTERRUPT ROUTINE
3200  017570  017704  162252          8$:     MOV     @DZRBUF,R4              ;ACTUAL
3201  017574  010403                          MOV     R4,R3
3202  017576  000303                          SWAB    R3
3203  017600  042703  177770                  BIC     #^C<7>,R3       ;STRIP JUNK
3204  017604  105737  001371                  TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MODE?
3205  017610  001406                          BEQ     11$             ;IF NOT, SKIP STAGGERED SETUP
3206
3207                                          ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
3208
3209  017612  006203                          ASR     R3              ;GET THE LAST BIT INTO THE CARRY BIT
3210  017614  103402                          BCS     9$              ;IF IT IS SET, GO CLEAR IT
3211  017616  000261                          SEC                     ;IF IT IS CLEAR SET IT HERE
3212  017620  000401                          BR      10$             ;SKIP THE CLEARING
3213  017622  000241                  9$:     CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
3214  017624  006103                  10$:    ROL     R3              ;GET THE NEW BIT BACK INTO R3
3215  017626  020337  001372          11$:    CMP     R3,SAVLIN       ;IS THIS A VALID LINE
3216  017632  001401                          BEQ     .+4             ;YES
3217  017634  104015                          ERROR   15              ;*INVALID LINE
3218  017636  042704  177400                  BIC     #^C<377>,R4     ;STRIP JUNK
3219  017642  120504                          CMPB    R5,R4           ;DATA COMPARE ?
3220  017644  001401                          BEQ     .+4             ;YES
3221  017646  104005                          ERROR   5               ;*DATA DOES NOT COMPARE
3222  017650  040277  162202                  BIC     R2,@DZTCR       ;CLEAR TCR BIT
3223  017654  022626                          POP2SP                  ;REMOVE HE INTERRUPT VECTOR FROM THE STACK
3224  017656  000402                          BR      13$             ;GO GET OUT OF INTERRUPT MODE
3225                                          ;TRANSMITTER INTERRUPT SVC ROUTINE
3226  017660  104011                  12$:    ERROR   11              ;THE RECEIVER INTERRUPT FAILED
3227                                          ;TO OVERRIDE THE TRANSMITTER
3228  017662  022626                          POP2SP                  ;REMOVE THE INTERRUPT VECTOR FROM THE STACK
```

```
3229  017664  042777  040100  162150  13$:    BIC     #TIE!RIE,@DZCSR ;CLEAR INTERRUPT ENABLES
3230  017672  013777  002074  162172          MOV     DZRIS,@DZRIV    ;RESTORE TRAPCATCHER
3231  017700  005077  162170                  CLR     @DZRIS
3232  017704  013777  002100  162164          MOV     DZTIS,@DZTIV
3233  017712  005077  162162                  CLR     @DZTIS
3234                                          ;************************ TEST 30 *******************************
3235                                          ;*TEST TO VERIFY THAT 'RDONE DOES NOT SET
3236                                          ;*IF THE SCANNER IS DISABLED.
3237                                          ;*TURN ON SCANNER, WAIT FOR TRDY,
3238                                          ;*TURN OFF SCANNER, TRANSMIT A CHARACTER
3239                                          ;*'RDONE SHOULD NOT SET.
3240                                  ;:*  TEST 30
3241                                  ;:*************************************************************
3242  017716  000004                  TST30:  SCOPE
3243  017720  012737  000030  001122          MOV     #30,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
3244  017726  012737  020104  001360          MOV     #TST31,NEXT     ;POINT TO THE START OF THE NEXT TEST
3245  017734  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3246  017736  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
3247  017742  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
3248  017746  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
3249  017752  001402                          BEQ     2$              ;NO
3250  017754  010177  162072                  MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
3251  017760  005201                  2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
3252  017762  106302                          ASLB    R2              ;GOT 'EM ALL ?
3253  017764  103370                          BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
3254  017766  005037  001372                  CLR     SAVLIN          ;CLEAR LINE # INDICATOR
3255  017772  052777  000040  162042          BIS     #MSENAB,@DZCSR  ;TURN ON SCANNER
3256  020000  012702  000001                  MOV     #1,     R2      ;INIT LINE COUNTER
3257  020004  030237  001364          3$:     BIT     R2,     LINE    ;FIND A VALID LINE
3258  020010  001004                          BNE     4$              ;IF WE FOUND ONE GO TO TEST
3259  020012  005237  001372                  INC     SAVLIN          ;IF NOT
3260  020016  106302                          ASLB    R2              ;KEEP LOOKING
3261  020020  000771                          BR      3$
3262  020022  110277  162030          4$:     MOVB    R2,     @DZTCR  ;SET TCR BIT
3263  020026  005005                          CLR     R5
3264  020030  005777  162006          5$:     TST     @DZCSR          ;IS TRDY SET
3265  020034  100404                          BMI     6$              ;CON'T TESTING IF IT IS
3266  020036  104414                          DELAY                   ;IF IT NOT WAIT A WHILE
3267  020040  005205                          INC     R5
3268  020042  001372                          BNE     5$
3269  020044  104003                          ERROR   3               ;WE WAITED LONG ENOUGH-ERROR
3270  020046  042777  000040  161766  6$:     BIC     #MSENAB, @DZCSR ;TURN OFF SCANNER
3271  020054  105077  162006                  CLRB    @DZTDR          ;TRANSMIT A CHARACTER
3272  020060  005005                          CLR     R5              ;CLEAR COUNTER
3273  020062  104414                  7$:     DELAY                   ;WAIT SUFFICIENT TIME FOR
3274  020064  005205                          INC     R5              ;RDONE TO SET
3275  020066  001375                          BNE     7$
3276  020070  032777  000200  161744          BIT     #RDONE, @DZCSR  ;RDONE SET
3277  020076  001401                          BEQ     8$              ;IT SHOULDN'T BE-CONTINUE
3278  020100  104020                          ERROR   20              ;IF IT IS THERE'S AN ERROR
3279  020102  104400                  8$:     ADVANCE
3280                                          ;************************ TEST 31 *******************************
3281                                          ;*THIS TEST VERIFIES OVERRUN AND SILO ALARM
3282                                          ;*ONE LINE AT A TIME  - BASED UPON VALID LINES
3283                                          ;*AS EACH OF THE FIRST 16 CHARS ARE SENT; SILO ALARM IS
3284                                          ;*TESTED TO BE CLEARED. ON THE 16TH CHAR THE PROGRAM THEN
```

```
                                        ;*EXPECTS SILO ALARM TO SET. THEN THE ENTIRE
                                        ;*SILO IS FILLED AND AN OVERRUN IS EXPECTED ON THE 65TH
                                        ;*CHAR PULLED OUT OUT THE SILO.
                                        ;*USING SWITCH NINE FOR THIS TEST SENDS 20. CHARACTERS
                                        ;*ON DZ LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
                                        ;*USED TO SCOPE SILO ALARM PULSES, ETC.
                                ;:*  TEST 31
                                ;:*****************************************************************
020104  000004                  TST31:  SCOPE
020106  012737  000031  001122          MOV     #31,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
020114  012737  020632  001360          MOV     #TST32,NEXT     ;POINT TO THE START OF THE NEXT TEST
020122  012737  020536  001362          MOV     #18$,LOCK       ;SET FOR LOOP
020130  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
020132  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
020136  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
020142  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
020146  001402                          BEQ     2$              ;NO
020150  010177  161676                  MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
020154  005201                  2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
020156  106302                          ASLB    R2              ;GOT 'EM ALL ?
020160  103370                          BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
020162  005037  001372                  CLR     SAVLIN          ;CLEAR LINE # INDICATOR
020166  012700  001422                  MOV     #TDO,R0         ;POINT TO THE DATA AREA
020172  005020                          CLR     (R0)+           ;CLEAR A DATA WORD
020174  022700  001462                  CMP     #STOP,R0        ;FINISHED ?
020200  001374                          BNE     .-6             ;NO
020202  005000                          CLR     R0              ;CLEAR OFFSET
020204  012702  000001                  MOV     #1,R2           ;LINE POINTER
020210  052777  010040  161624          BIS     #MSENAB!SILOEN,@DZCSR   ;START SCANNER & SET SILO ENABLE
020216  030237  001364          3$:     BIT     R2,LINE         ;VALID LINE?
020222  001002                          BNE     .+6             ;YES
020224  000137  020520                  JMP     22$             ;TRY NEXT LINE
020230  013700  001372                  MOV     SAVLIN,R0       ;MAKE OFFSET
020234  006300                          ASL     R0              ;MAKE POWER OF TWO
020236  010277  161614                  MOV     R2,@DZTCR       ;SET TCR BIT
020242  105777  161574          4$:     TSTB    @DZCSR          ;REC DONE = 1 ?
020246  100001                          BPL     .+4
020250  104020                          ERROR   20              ;REC DONE SHOULD NOT = 1
020252  005003                          CLR     R3              ;SET CHARACTER COUNT
020254  005004                  5$:     CLR     R4
020256  032777  100000  161556  6$:     BIT     #TRDY,@DZCSR
020264  001004                          BNE     7$
020266  104414                          DELAY
020270  105204                          INCB    R4
020272  001371                          BNE     6$
020274  104003                          ERROR   3               ;*TRDY FAILED TO SET
020276  116077  001422  161562  7$:     MOVB    TDO(R0),@DZTDR  ;LOAD A CHARACTER
020304  005260  001422                  INC     TDO(R0)         ;SET UP NEXT CHARACTER
020310  020327  000017                  CMP     R3,#15.         ;16 CHARACTERS ?
020314  103006                          BHIS    8$
020316  032777  020000  161516          BIT     #SILOAL,@DZCSR  ;SILO ALARM = 0 ?
020324  001401                          BEQ     .+4             ;YES
020326  104013                          ERROR   13              ;*SILO ALARM SHOULD NOT = 1
                                                                ;UNTIL 16. DATA CHARACTERS
020330  000411                          BR      10$
020332  005004                  8$:     CLR     R4
```

```
3341  020334  032777  020000  161500  9$:     BIT     #SILOAL,@DZCSR
3342  020342  001004                          BNE     10$
3343  020344  104414                          DELAY
3344  020346  005204                          INC     R4
3345  020350  001371                          BNE     9$
3346  020352  104014                          ERROR   14                ;*SILO ALARM FAILED TO SET!
3347                                                                    ;SILO ALARM SHOULD =1 AFTER 16.
3348                                                                    ;DATA CHARACTERS
3349  020354  005203                  10$:    INC     R3                ;INC CHAR COUNT
3350  020356  022703  000102                  CMP     #66.,R3           ;FINISHED SENDING CHARACTERS ?
3351  020362  001334                          BNE     5$                ;NO
3352  020364  005004                          CLR     R4
3353  020366  104414                          DELAY
3354  020370  105204                          INCB    R4
3355  020372  001375                          BNE     .-4
3356                                          ;NOW LETS READ THE SILO
3357  020374  013705  001372                  MOV     SAVLIN,R5         ;MAKE EXPECTED LINE #
3358  020400  105737  001371                  TSTB    MODE+1            ;IS THIS TEST IN STAGGERED MODE?
3359  020404  001406                          BEQ     13$               ;IF NOT, SKIP STAGGERED SETUP
3360
3361                                          ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
3362
3363  020406  006205                          ASR     R5                ;GET THE LAST BIT INTO THE CARRY BIT
3364  020410  103402                          BCS     11$               ;IF IT IS SET, GO CLEAR IT
3365  020412  000261                          SEC                       ;IF IT IS CLEAR SET IT HERE
3366  020414  000401                          BR      12$               ;SKIP THE CLEARING
3367  020416  000241                  11$:    CLC                       ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
3368  020420  006105                  12$:    ROL     R5                ;GET THE NEW BIT BACK INTO R5
3369  020422  000305                  13$:    SWAB    R5                ;PUT IN UPPER BYTE
3370  020424  052705  100000                  BIS     #DVALID,R5        ;ADD DATA VALID
3371  020430  017704  161412          14$:    MOV     @DZRBUF,R4        ;ACTUAL
3372  020434  020405                          CMP     R4,R5             ;ACTUAL VS. EXPECTED
3373  020436  001401                          BEQ     15$               ;YES
3374  020440  104006                          ERROR   6                 ;*DATA/CONTENTS DID NOT COMPARE
3375  020442  032777  020000  161372  15$:    BIT     #SILOAL,@DZCSR    ;SILO ALARM= 0 ?
3376  020450  001401                          BEQ     16$               ;YES
3377  020452  104016                          ERROR   16                ;READING DZRBUF DID NOT CLEAR SILO ALARM
3378  020454  005205                  16$:    INC     R5                ;UP CHARACTER
3379  020456  120527  000077                  CMPB    R5,#63.           ;LAST SILO CHAR ?....64TH CHAR
3380  020462  101762                          BLOS    14$
3381  020464  005205                          INC     R5                ;ADD 1 MORE FOR THE CLOBBERED CHAR
3382  020466  052705  040000                  BIS     #OVRRUN,R5        ;ADD OVERRUN TO EXPECTED
3383  020472  120527  000101                  CMPB    R5,#65.           ;LAST CHARACTER ?
3384  020476  001754                          BEQ     14$
3385  020500  017704  161342                  MOV     @DZRBUF,R4        ;FOR GOOD MEASURE
3386  020504  005704                          TST     R4                ;DATA VALID SHOULD = 0
3387  020506  100001                          BPL     17$               ;YES
3388  020510  104017                          ERROR   17                ;DATA VALID SHOULD = 0
3389  020512  040277  161340          17$:    BIC     R2,@DZTCR         ;CLR TCR BIT
3390  020516  104401                          SCOP1                     ;LOOP?
3391  020520  005237  001372          22$:    INC     SAVLIN            ;INC EXPECTED LINE
3392  020524  106302                          ASLB    R2                ;NEXT LINE
3393  020526  103402                          BCS     .+6               ;NO
3394  020530  000137  020216                  JMP     3$                ;YES
3395  020534  104400                          ADVANCE                   ;GO TO NEXT TEST
3396
```

```
                                        ;TIGHT SCOPE LOOP FOR THIS TEST. SENDS 20. CHARACTERS
                                        ;ON DZ LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
                                        ;USED TO SCOPE SILO ALARM PULSES, ETC.

020536  052777  010040  161276  18$:    BIS     #MSENAB!SILOEN,@DZCSR   ;SETUP DEVICE
020544  012777  020622  161324          MOV     #20$,@DZTIV             ;SETUP TRANSMITTER VECTOR
020552  012737  000024  001216          MOV     #20.,$TMPO              ;TEMPORARY COUNT OF CHARACTER BURST
020560  050277  161272                  BIS     R2,@DZTCR               ;ENABLE LINE
020564  052777  040000  161250          BIS     #TIE,@DZCSR             ;ENABLE INTERRUPTS
020572  106427  000000                  MTPS    #0                      ;LOWER PRIORITY
020576  000001                  19$:     WAIT                            ;ALLOW INTERRUPTS
020600  005337  001216                  DEC     $TMPO                   ;REDUCE COUNT. ALL CHARACTERS SENT?
020604  001374                          BNE     19$                     ;IF NO, WAIT FOR MORE
020606  042777  050040  161226          BIC     #SILOEN!MSENAB!TIE,@DZCSR ;RESET SILO COUNTER, CLEAR STROBE
020614  104401                          SCOP1                           ;LOOP AGAIN?
020616  000137  020512                  JMP     17$                     ;IF NOT, RETURN TO WHERE YOU LEFT OFF
020622  112777  000252  161236  20$:    MOVB    #252,@DZTDR             ;SEND A CHARACTER
020630  000002                          RTI                             ;ALLOW MORE CHARACTERS TO COME
                                        ;********************** TEST 32 ******************************
                                        ;*THIS TEST THAT "SILO ENABLE" WILL INHIBIT
                                        ;*RECEIVER INTERRUPTS AND THAT ON THE
                                        ;*16TH CHAR THAT "SILO ALARM" WILL CAUSE AN
                                        ;*INTERRUPT WITH "RIE" SET.
                                        ;*THIS WILL DO ALL SELECTED LINES ONE AT A TIME.
                                ;:*  TEST 32
                                ;:*************************************************************
020632  000004                  TST32:  SCOPE
020634  012737  000032  001122          MOV     #32,$TSTNM              ;LOAD THE NUMBER OF THIS TEST
020642  012737  021214  001360          MOV     #TST33,NEXT             ;POINT TO THE START OF THE NEXT TEST
020650  012737  020736  001362          MOV     #3$,LOCK                ;SET FOR LOOP
020656  104417                          DCLASM                          ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
020660  013701  001366                  MOV     PAR,R1                  ;PICK UP PARAMETERS
020664  012702  000001                  MOV     #1,R2                   ;PICK UP INIT POINTER
020670  030237  001364          1$:     BIT     R2,LINE                 ;SHOULD THIS LINE BE SET UP ?
020674  001402                          BEQ     2$                      ;NO
020676  010177  161150                  MOV     R1,@DZLPR               ;SET UP LINE PARAMETERS
020702  005201                  2$:     INC     R1                      ;POSITION POINTER TO THE NEXT LINE
020704  106302                          ASLB    R2                      ;GOT 'EM ALL ?
020706  103370                          BCC     1$                      ;IF NO, GO SET UP THE NEXT LINE
020710  005037  001372                  CLR     SAVLIN                  ;CLEAR LINE # INDICATOR
020714  012700  001422                  MOV     #TDO,R0                 ;POINT TO THE DATA AREA
020720  005020                          CLR     (R0)+                   ;CLEAR A DATA WORD
020722  022700  001462                  CMP     #STOP,R0                ;FINISHED ?
020726  001374                          BNE     .-6                     ;NO
020730  005000                          CLR     R0                      ;CLEAR OFFSET
020732  012702  000001                  MOV     #1,R2                   ;LINE POINTER
020736  012777  021156  161126  3$:     MOV     #11$,@DZRIV             ;SET FOR UNEXPECTED INTER.
020744  012777  000340  161122          MOV     #PR7,@DZRIS             ;SET PRIO.
020752  052777  010140  161062          BIS     #MSENAB!SILOEN!RIE,@DZCSR
                                                                        ;START SCANNER & SET SILO ENABLE
020760  030237  001364                  BIT     R2,LINE                 ;VALID LINE?
020764  001002                          BNE     .+6                     ;YES
020766  000137  021174                  JMP     22$                     ;TRY NEXT LINE
020772  005777  161050                  TST     @DZRBUF                 ;EMPTY THE SILO
020776  100775                          BMI     .-4                     ;BR IF DATA VALID IS SET!
021000  106427  000000                  MTPS    #0                      ;SET PROCESSOR PRIORITY TO 0
```

```
3453  021004  013700  001372                    MOV     SAVLIN,R0       ;MAKE OFFSET
3454  021010  006300                            ASL     R0              ;MAKE POWER OF TWO
3455  021012  010277  161040                    MOV     R2,@DZTCR       ;SET TCR BIT
3456  021016  005004                    5$:     CLR     R4
3457  021020  032777  100000  161014    6$:     BIT     #TRDY,@DZCSR
3458  021026  001004                            BNE     7$
3459  021030  104414                            DELAY
3460  021032  005204                            INC     R4
3461  021034  001371                            BNE     6$
3462  021036  104003                            ERROR   3               ;*TRDY FAILED TO SET
3463  021040  116077  001422  161020  7$:      MOVB    TDO(R0),@DZTDR  ;LOAD A CHARACTER
3464  021046  005260  001422                    INC     TDO(R0)         ;SET UP NEXT CHARACTER
3465  021052  022760  000017  001422            CMP     #15.,TDO(R0)    ;15 CHARS YET?
3466  021060  001406                            BEQ     8$
3467  021062  032777  020000  160752            BIT     #SILOAL,@DZCSR  ;SILO ALARM = 0 ?
3468  021070  001401                            BEQ     .+4             ;YES
3469  021072  104013                            ERROR   13              ;*SILO ALARM SHOULD NOT = 1
3470                                                                    ;UNTIL 16. DATA CHARACTERS
3471  021074  000750                            BR      5$
3472  021076  012777  021164  160766  8$:      MOV     #12$,@DZRIV     ;SET NEW VECTOR
3473  021104  032777  100000  160730            BIT     #TRDY,@DZCSR    ;READY FOR 16TH CHAR
3474  021112  001774                            BEQ     .-6
3475  021114  016077  001422  160744            MOV     TDO(R0),@DZTDR  ;LOAD THE 16TH CHAR.
3476  021122  005004                            CLR     R4
3477  021124  032777  020000  160710  9$:      BIT     #SILOAL,@DZCSR
3478  021132  001005                            BNE     10$
3479  021134  104414                            DELAY
3480  021136  005204                            INC     R4
3481  021140  001371                            BNE     9$
3482  021142  104014                            ERROR   14              ;*SILO ALARM FAILED TO SET!
3483  021144  000410                            BR      17$             ;SILO ALARM SHOULD =1 AFTER 16.
3484                                                                    ;DATA CHARACTERS
3485  021146  000240                    10$:    NOP                     ;STALL
3486  021150  000240                            NOP
3487  021152  104000                            ERROR                   ;SILO ALARM NOT INTERRUPTING.
3488  021154  000404                            BR      17$             ;CONTINUE TEST.
3489  021156  022626                    11$:    CMP     (SP)+,(SP)+     ;FAKE RTI
3490  021160  104012                            ERROR   12              ;RX SHOULD NOT INTERRUPT
3491  021162  000401                            BR      17$             ;CONTINUE
3492  021164  022626                    12$:    CMP     (SP)+,(SP)+     ;GOOD INTERRUPT TO HERE.
3493  021166  040277  160664          17$:      BIC     R2,@DZTCR       ;CLR TCR BIT
3494  021172  104401                            SCOP1                   ;LOOP?
3495  021174  005237  001372          22$:      INC     SAVLIN          ;INC EXPECTED LINE
3496  021200  106302                            ASLB    R2              ;NEXT LINE
3497  021202  103402                            BCS     .+6             ;NO
3498  021204  000137  020736                    JMP     3$              ;YES
3499  021210  005037  001362                    CLR     LOCK            ;CLEAR TIGHT LOOP FOR NEXT TEST
```

```
                                        ;*****************  TEST 33 ******************************
                                        ;*THIS TEST RUNS ALL LINES FULL BORE
                                        ;*BASED UPON QUALIFIED LINES
                                        ;*..THIS IS AN INTERRUPT TEST ON THE RECEIVER AND
                                        ;*TRANSMITTER
                                ;:*  TEST 33
                                ;:*************************************************************
021214  000004                  ↑ST33:  SCOPE
021216  012737  000033  001122          MOV     #33,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
021224  012737  022022  001360          MOV     #↑ST34,NEXT     ;POINT TO THE START OF THE NEXT TEST
021232  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
021234  013737  001364  022020          MOV     LINE,RXTCR      ;SET IMAGE OF TCR BITS
021242  013701  001366          RSTART: MOV     PAR,R1          ;PICK UP PARAMETER
021246  012700  000001                  MOV     #1,R0           ;.PICK UP INIT POINTER
021252  030037  001364          INIT:   BIT     R0,LINE         ;SHOULD THIS LINE BE SET UP
021256  001402                          BEQ     1$              ;NO
021260  010177  160566                  MOV     R1,@DZLPR       ;SET UP LINE PARAM REGISTER
021264  005201                  1$:     INC     R1
021266  106300                          ASLB    R0              ;GOT 'EM ALL ?
021270  103370                          BCC     INIT            ;NO
021272  012700  001422                  MOV     #TDO,R0         ;CLEAR TRANS DATA POINTER & REC POINTERS
021276  005020                  INIT1:  CLR     (R0)↓
021300  022700  001462                  CMP     #STOP,R0                ;FINISHED ?
021304  001374                          BNE     INIT1           ;NO, CONTINUE CLEARING
021306  012777  021542  160556          MOV     #RXSVC,@DZRIV   ;SET UP REC INTR VECTOR
021314  012777  000340  160552          MOV     #PR7,@DZRIS     ;STATUS
021322  012777  021444  160546          MOV     #↑XSVC,@DZTIV   ;SET UP TRANS INTR VECTOR
021330  012777  000340  160542          MOV     #PR7,@DZTIS     ;STATUS
021336  052777  000100  160476          BIS     #RIE,@DZCSR             ;SET REC INTR ENABLE
021344  052777  040000  160470          BIS     #TIE,@DZCSR             ;SET TRANS INTR ENABLE
021352  052777  000040  160462          BIS     #MSENAB,@DZCSR  ;SET MASTER SCAN ENABLE
021360  113777  001364  160470          MOVB    LINE,@DZTCR     ;SET TCR BITS...UP UP AND AWAY !
021366  106437  027062                  MTPS    @#LESS1                 ;ALLOW INTERRUPTS


021372  005037  021442          SNAP:   CLR     66$
021376  013727  006722          67$:    MOV     DLYCNT,(PC)+    ;SET FOR DELAY
021402  000000          68$:    0
021404  005337  021402                  DEC     68$
021410  001375                          BNE     .-4
021412  105737  022020                  TSTB    RXTCR           ;WAIT FOR ALL RECIEVERS TO FINISH
021416  001002                          BNE     3$              ;
021420  000137  021720                  JMP     OUT
021424  005237  021442          3$:     INC     66$
021430  001362                          BNE     67$
021432  104007                          ERROR   7               ;*TRANSMITTER FAILED TO INTERRUPT
021434  104011                          ERROR   11              ;*RECEIVER FAILED TO INTERRUPT
021436  000137  021772                  JMP     FINI
021442  000000          66$:    0

                                ;TRANS INTR SVC ROUTINE
021444  005777  160372          ↑XSVC:  TST     @DZCSR          ;TRANS INTR ?
021450  100401                          BMI     .+4
021452  104003                          ERROR   3               ;*TRANSMITTER FAILED
021454  117703  160364                  MOVB    @HDZCSR,R3      ;SAVE IT
                                ;NOW TEST FOR LINE # ETC
```

```
3556  021460  042703  177770            BIC     #^C<7>,R3       ;STRIP JUNK
3557  021464  010304                    MOV     R3,R4           ;SAVE
3558  021466  010337  001372            MOV     R3,SAVLIN       ;ADJUST LOCATION FOR ERROR PRINTOUT
3559  021472  012702  000001            MOV     #1,R2           ;SET UP POSITION POINTER
3560  021476  105303            3$:     DECB    R3              ;IS IT THIS LINE ?
3561  021500  100402                    BMI     4$              ;YES
3562  021502  006302                    ASL     R2              ;UP THE LINE #
3563  021504  000774                    BR      3$              ;GO 'ROUND AGAIN
3564  021506  030237  001364      4$:   BIT     R2,LINE         ;VALID LINE?
3565  021512  001001                    BNE     .+4             ;YES
3566  021514  104010                    ERROR   10              ;NO,INVALID LINE!!!!
3567  021516  006304                    ASL     R4              ;MAKE POWER OF 2
3568  021520  116477  001422  160340    MOVB    TDO(R4),@DZTDR  ;LOAD CHARACTER
3569  021526  105264  001422            INCB    TDO(R4)         ;SET UP NEXT CHARACTER
3570  021532  001002                    BNE     5$              ;LAST CHARACTER ?
3571  021534  040277  160316            BIC     R2,@DZTCR       ;YES ,CLEAR TCR BIT
3572  021540  000002            5$:     RTI
3573
3574
3575                              ;REC INTR SVC ROUTINE
3576  021542  105777  160274    RXSVC:  TSTB    @DZCSR          ;REC DONE ?
3577  021546  100401                    BMI     .+4             ;YES
3578  021550  104004                    ERROR   4               ;FALSE INTERRUPT
3579  021552  017704  160270            MOV     @DZRBUF,R4      ;SAVE IT
3580  021556  010403                    MOV     R4,R3
3581  021560  000303                    SWAB    R3
3582  021562  042703  177770            BIC     #^C<7>,R3       ;STRIP JUNK
3583  021566  010337  001372            MOV     R3,SAVLIN       ;SAVE LINE NUMBER
3584  021572  032777  020000  160242    BIT     #SILOAL,@DZCSR  ;SILO ALARM?
3585  021600  001401                    BEQ     .+4             ;NO
3586  021602  104000                    ERROR                   ;SILO ALARM SHOULD NOT =1
3587  021604  005704                    TST     R4              ;DATA VALID SET?
3588  021606  100401                    BMI     .+4             ;YES
3589  021610  104023                    ERROR   23              ;YOU LOSE  ...DATA VALID WAS'NT SET
3590  021612  032704  070000            BIT     #OVRRUN!FRMERR!PARER,R4
3591  021616  001401                    BEQ     .+4
3592  021620  104000                    ERROR                   ;RECEIVER ERROR FLAG/S WERE SET
3593  021622  012702  000001            MOV     #1,R2           ;SET UP POSITION POINTER
3594  021626  105303            5$:     DECB    R3
3595  021630  100402                    BMI     6$
3596  021632  006302                    ASL     R2              ;RE POSITION POINTER
3597  021634  000774                    BR      5$              ;GO 'ROUND AGAIN
3598  021636  030237  001364      6$:   BIT     R2,LINE         ;LINE VALID ?
3599  021642  001001                    BNE     .+4             ;YES
3600  021644  104011                    ERROR   11              ;INVALID LINE #
3601  021646  013703  001372            MOV     SAVLIN,R3       ;GET THE LINE NUMBER AGAIN
3602  021652  006303                    ASL     R3              ;USE R3 AS A POINTER IN THE DATA TABLE
3603  021654  126304  001442            CMPB    TRO(R3),R4      ;DOES THE DATA CHARACTER COMPARE ?
3604  021660  001405                    BEQ     2$              ;YES
3605  021662  016305  001442            MOV     TRO(R3),R5      ;SAVE EXPECTED
3606  021666  042704  177400            BIC     #^C<377>,R4     ;CLEAR JUNK
3607                                                            ;R2 = LINE # BY BIT POSITION
3608                                                            ;R4 = ACTUAL DATA
3609                                                            ;R5 = EXPECTED DATA
3610  021672  104005                    ERROR   5               ;*NO, DATA DOES NOT COMPARE
3611  021674  005263  001442      2$:   INC     TRO(R3)         ;SET UP FOR NEXT CHARACTER
```

```
3612  021700  105763  001442                TSTB    TRO(R3) ;ALL CHARS DONE?
3613  021704  001002                        BNE     .+6
3614  021706  040237  022020                BIC     R2,RXTCP        ;ZERO LINE DONE INDICATOR.
3615  021712  012716  021372                MOV     #SNAP,(SP)      ;RESET THE BACKGROUND TIMING LOOP
3616  021716  000002                        RTI
3617
3618
3619                                        ;FINISH UP ROUTINE
3620  021720  106427  000340          OUT:  MTPS    #PR7                    ;STOP ALL INTERRUPTS
3621  021724  104413                        DEVICE.CLR                    ;CLEAR ALL INTERRUPTS AWAY
3622  021726  005003                        CLR     R3
3623  021730  005037  001372                CLR     SAVLIN
3624  021734  012702  000001                MOV     #1,R2
3625  021740  030237  001364          1$:   BIT     R2,LINE         ;VALID LINE ?
3626  021744  001405                        BEQ     2$              ;NO
3627  021746  022763  000400  001442        CMP     #400,TRO(R3)    ;RECEIVED A BINARY COUNT PATTERN ?
3628  021754  001401                        BEQ     .+4             ;YES
3629  021756  104027                        ERROR   27              ;THE LINE FAILED TO RECEIVE A FULL
3630                                                                ;BINARY COUNT PATTERN
3631  021760  005237  001372          2$:   INC     SAVLIN          ;SET UP FOR NEXT LINE
3632  021764  005723                        TST     (R3)+           ;ADD 2
3633  021766  106302                        ASLB    R2              ;SET UP NEXT LINE POINTER
3634  021770  103363                        BCC     1$              ;FINISHED ?
3635  021772                          FINI:
3636  021772  013777  002074  160072        MOV     DZRIS,@DZRIV    ;RESTORE TRAPCATCHER
3637  022000  005077  160070                CLR     @DZRIS
3638  022004  013777  002100  160064        MOV     DZTIS,@DZTIV
3639  022012  005077  160062                CLR     @DZTIS
3640  022016  104400                        ADVANCE                 ;GO TO THE NEXT TEST
3641  022020  000000                  RXTCR: 0                       ;RX IMAGE OF TCR BITS
3642
3643
3644                                        ;************************ TEST 34 *******************************
3645                                        ;*DZ11 RELATIVE TIMING TEST.
3646                                        ;*EACH SELECTED LINE WILL IN TURN RUN 16. CHARS
3647                                        ;*AT ALL BAUD RATES AND THEN  E HIGHEST BAUD
3648                                        ;*WITH ALL CHAR LENGTHS. EACH NEW PARAMETER SHOULD
3649                                        ;*DECREASE IN TIME FROM THE PREVIOUS PARAMETERS SELECTED.
3650                                        ;*THE TIME IS CHECKED AGAINST THE LAST PARAMETER USED
3651                                        ;* AND A LOWER TIME IS EXPECTED ON THE CURRENT PARAMETER.
3652                                        ;*PARAMETERS ARE:
3653                                        ;*  EIGHT BITS/PER/CHAR - TWO STOP BITS AT
3654                                        ;*       50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000
3655                                        ;*       2400, 3600, 4800, 7200, 9600 BAUD.
3656                                        ;*  19.2 K BAUD - TWO STOP BITS AT
3657                                        ;*       SEVEN, SIX, FIVE BITS/PER/CHAR.
3658                                        ;*AFTER EACH LINE HAS FINISHED ALL THE ABOVE PARAMETERS
3659                                        ;*THE NEXT SELECTED LINE IS THE TESTED.
3660                                  ;:*  TEST 34
3661                                  ;:************************************** **********************
3662  022022  000004                 TST34: SCOPE
3663  022024  012737  000034  001122        MOV     #34,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
3664  022032  012737  000002  001226        MOV     #2,$TIMES
3665  022040  012737  022532  001360        MOV     #TST35,NEXT     ;POINT TO THE START OF THE NEXT TEST
3666  022046  012737  022172  001362        MOV     #3$,LOCK        ;SET FOR LOOP
3667  022054  005037  023766                CLR     OFFSET          ;RESET THIS VARIABLE
```

```
 3668  022060  005037  001372                    CLR     SAVLIN          ;RESET LINE NUMBER INDICATOR
 3669  022064  005037  001374                    CLR     XMTLIN          ;USE THIS WORD TO TELL WHAT LINE TRANSMITTED
 3670  022070  012737  000001  001216            MOV     #1,$TMPC        ;USE $TMPC AS A BIT POINTER
 3671  022076  012737  010070  022530            MOV     #RCVON!SSC!EIGHT!TWOSTOP,7$ ;BUILD TEMPORARY PARAMETERS
 3672  022104  033737  001216  001364  1$:       BIT     $TMPC,LINE      ;IS THIS LINE ACTIVE?
 3673  022112  001027                            BNE     3$              ;IF SO, GO GET STARTED
 3674  022114  012737  010070  022530  2$:       MOV     #RCVON!SSC!EIGHT!TWOSTOP,7$ ;LOAD PARAMETERS TEMPORARILY
 3675  022122  012700  001422                    MOV     #TDC,R0         ;POINT TO THE DATA AREA
 3676  022126  005020                            CLR     (R0)+           ;CLEAR A DATA WORD
 3677  022130  022700  001462                    CMP     #STOP,R0        ;FINISHED ?
 3678  022134  001374                            BNE     .-6             ;NO
 3679  022136  005237  001374                    INC     XMTLIN          ;POINT TO THE NEXT LINE TO TRANSMIT
 3680  022142  042737  000007  022530            BIC     #7,7$           ;MAKE SURE TEMPORARY PARAMETERS POINT TO 0
 3681  022150  053737  001374  022530            BIS     XMTLIN,7$       ;ADD DESIRED LINE NUMBER
 3682  022156  005037  023766                    CLR     OFFSET
 3683  022162  106337  001216                    ASLB    $TMPC           ;POINT TO THE NEXT LINE
 3684  022166  103346                            BCC     1$              ;PROCESS THE NEXT LINE
 3685  022170  104400                            ADVANCE                 ;TEST TO SEE IF THIS TEST GETS REPEATED
 3686  022172                          3$:
 3687  022172  104417                            DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
 3688  022174  042737  010000  022530            BIC     #RCVON,7$       ;ZERO PARAMTERS FOR TX LINE
 3689  022202  013777  022530  157642            MOV     7$,@DZLPR       ;LOAD PARAMTERS FOR TX
 3690  022210  005737  001370                    TST     MODE            ;STAGGERED?
 3691  022214  100011                            BPL     100$            ;BR IF NO
 3692  022216  000241                            CLC                     ;SET UP LINE
 3693  022220  006037  022530                    ROR     7$              ;
 3694  022224  103002                            BCC     98$             ;BR IF LINE WAS EVEN
 3695  022226  000241                            CLC                     ;PREPARE TO MKE LINE EVEN
 3696  022230  000401                            BR      99$             ;CONTINUE
 3697  022232  000261                  98$:      SEC                     ;PREPARE TO MAKE LINE ODD
 3698  022234  006137  022530          99$:      ROL     7$              ;SET ALTERED LINE
 3699  022240  052737  010000  022530  100$:     BIS     #RCVON,7$       ;SET RX ON
 3700  022246  013777  022530  157576            MOV     7$,@DZLPR       ;LOAD RX PARAMETERS
 3701  022254  013737  022530  001372            MOV     7$,SAVLIN       ;ADJUST LOCATION FOR ERROR PRINTOUT
 3702  022262  042737  177770  001372            BIC     #^C<7>,SAVLIN   ;STRIP JUNK
 3703  022270  042737  000007  022530            BIC     #7,7$           ;CLEAR OLD LINE #
 3704  022276  053737  001374  022530            BIS     XMTLIN,7$       ;SET LINE UP AGAIN
 3705  022304  013737  022530  001400            MOV     7$,REGIST       ;SAVE PARAMETERS FOR PRINTOUT
 3706  022312  012700  001422                    MOV     #TDC,R0         ;POINT TO THE DATA AREA
 3707  022316  005020                            CLR     (R0)+           ;CLEAR A DATA WORD
 3708  022320  022700  001462                    CMP     #STOP,R0        ;FINISHED ?
 3709  022324  001374                            BNE     .-6             ;NO
 3710  022326  005002                            CLR     R2              ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
 3711  022330  005003                            CLR     R3              ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
 3712  022332  005037  001220                    CLR     $TMP1           ;INITIALIZE THE TIMER
 3713  022336  005037  001224                    CLR     $TMP3           ;INITIALIZE THESE BITS ALSO
 3714  022342  012737  000020  001376            MOV     #20,XMTCNT      ;SET HOW MANY CHARACTERS TO TRANSMIT
 3715  022350  012777  023410  157520            MOV     #XMTSRV,@DZTIV
 3716  022356  012777  023554  157506            MOV     #RXISR1,@DZRIV
 3717  022364  013777  027060  157502            MOV     DZPRT,@DZRIS
 3718  022372  013777  027060  157500            MOV     DZPRT,@DZTIS
 3719  022400  113777  001216  157450            MOVB    $TMPC,@DZTCR    ;START THE VALID LINE
 3720  022406  052777  040140  157426            BIS     #TIE!RIE!MSENAB,@DZCSR
 3721  022414  106427  000000                    MTPS    #0              ;LOWER THE PRIORITY TO ALLOW INTERRUPTS
 3722  022420  032777  000100  157414  4$:       BIT     #RIE,@DZCSR     ;IS ROUTINE DONE?
 3723  022426  001407                            BEQ     5$              ;WHEN ALL IS DONE RX IE IS CLEARED IN ISR.
```

```
3724  022430  005237  001220                  INC     $TMP1           ;COUNT TIME
3725  022434  001371                          BNE     4$              ;CONTINUE TEST
3726  022436  105237  001224                  INCB    $TMP3           ;DOUBLE COUNT
3727  022442  001366                          BNE     4$              ;CONTINUE TEST
3728  022444  104011                          ERROR   11              ;INTERRUPTS NOT FINISHED
3729  022446  004737  007360          5$:     JSR     PC,SERV.G       ;(↑G)?
3730  022452  104401                          SCOP1                   ;LOOP?
3731  022454  062737  000002  023766          ADD     #2,OFFSET
3732  022462  013700  022530                  MOV     7$,R0
3733  022466  042700  170377                  BIC     #↑C<17*400>,R0
3734  022472  022700  007400                  CMP     #<17*400>,R0
3735  022476  001010                          BNE     6$
3736  022500  032737  000030  022530          BIT     #BIT4+BIT3,7$
3737  022506  001602                          BEQ     2$
3738  022510  162737  000010  022530          SUB     #BIT3,7$
3739  022516  000625                          BR      3$
3740  022520  062737  000400  022530  6$:     ADD     #400,7$
3741  022526  000621                          BR      3$
3742  022530  000000                  7$:     0
3743                                          ;************************ TEST 35 *******************************
3744                                          ;* THIS TEST VERIFIES THAT EVEN PARITY WORKS
3745                                          ;* FOR ALL ODD LINES SELECTED AND THAT ODD PARITY WORKS FOR ALL
3746                                          ;* EVEN LINES SELECTED.
3747                                          ;*THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
3748                                          ;*THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
3749                                          ;*THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
3750                                          ;*YOU ARE IN "STAGGERED" MODE.
3751                                          ;*40(8) CHARS ARE USED FOR THIS TEST.
3752                                          ;*ALL SELECTED LINES WILL BE ENABLED
3753                                          ;*AT THE SAME TIME!
3754                                  ;:*  TEST 35
3755                                  ;:*****************************************************************
3756  022532  000004                  ↑ST35:  SCOPE
3757  022534  012737  000035  001122          MOV     #35,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
3758  022542  012737  022772  001360          MOV     #ST↑36,NEXT     ;POINT TO THE START OF THE NEXT TEST
3759  022550  005737  001370                  TST     MODE            ;IS THIS STAGGERED MODE?
3760  022554  100105                          BPL     6$              ;IF NOT, DON'T DO THIS TEST
3761  022556  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3762  022560  013701  001366                  MOV     PAR,R1          ;USE R1 TO BUILD PARAMETERS TO BE LOADED
3763  022564  042701  000200                  BIC     #ODDPAR,R1      ;MAKE SURE ODD PARITY ISN'T SET
3764  022570  052701  000100                  BIS     #PARITY,R1      ;MAKE SURE PARITY IS TURNED ON
3765  022574  012702  000001                  MOV     #1,R2           ;USE R2 AS A LINE POINTER
3766  022600  030237  001364          1$:     BIT     R2,LINE         ;IS THIS A VALID LINE?
3767  022604  001411                          BEQ     3$              ;IF NOT, SKIP TO THE NEXT LINE
3768  022606  032701  000001                  BIT     #BIT0,R1        ;IS THIS LINE AN ODD LINE?
3769  022612  001002                          BNE     2$              ;IF IT'S ODD, USE EVEN PARITY
3770  022614  052701  000200                  BIS     #ODDPAR,R1      ;IF IT'S EVEN, USE ODD PARITY
3771  022620  010177  157226          2$:     MOV     R1,@DZLPR       ;LOAD THE LINE PARAMETER REGISTER
3772  022624  042701  000200                  BIC     #ODDPAR,R1      ;SET UP THE NEXT PARITY TO EVEN
3773  022630  005201                  3$:     INC     R1              ;POINT TO THE NEXT LINE
3774  022632  106302                          ASLB    R2              ;MOVE THE BIT POINTER IN R2 TO THE NEXT LINE
3775  022634  103361                          BCC     1$              ;IF WE'RE NOT DONE, GO CHECK THE NEXT LINE
3776  022636  005037  001372                  CLR     SAVLIN          ;CLEAR THE LINE NUMBER INDICATOR
3777  022642  005002                          CLR     R2              ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
3778  022644  005003                          CLR     R3              ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
3779  022646  012737  000040  001376          MOV     #40,XMTCNT      ;TRANSMIT A BINARY COUNT PATTERN(00-40)
```

```
3780  022654  012700  001422                    MOV     #TDO,R0         ;POINT TO THE DATA AREA
3781  022660  005020                            CLR     (R0)+           ;CLEAR A DATA WORD
3782  022662  022700  001462                    CMP     #STOP,R0        ;FINISHED ?
3783  022666  001374                            BNE     .-6             ;NO
3784  022670  005000                            CLR     R0              ;CLEAR OFFSET
3785  022672  012777  023410  157176            MOV     #XMTSRV,@DZTIV  ;SET UP THE TRANSMITTER INTERRUPT VECTOR
3786  022700  012777  023232  157164            MOV     #PARESE,@DZRIV  ;SET UP THE RECEIVER INTERRUPT VECTOR
3787  022706  013777  027060  157160            MOV     DZPRT,@DZRIS    ;SET THE INTERRUPT VECTOR STATUS
3788  022714  013777  027060  157156            MOV     DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
3789  022722  052777  040140  157112            BIS     #RIE!TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
3790  022730  113777  001364  157120            MOVB    LINE,@DZTCR     ;ENABLE ALL SELECTED LINES
3791  022736  106427  000000                    MTPS    #0              ;ALLOW INTERRUPTS
3792  022742  032777  000100  157072  5$:       BIT     #RIE,@DZCSR     ;WHEN RX DONE; RIE WILL =0
3793  022750  001407                            BEQ     6$              ;BR IF ALL DONE
3794  022752  005237  023404                    INC     COUNT0
3795  022756  102771                            BVS     5$
3796  022760  105237  023406                    INCB    COUNT1
3797  022764  100366                            BPL     5$
3798  022766  104011                            ERROR   11              ;*RX FAILED TO FINISH (INTERRUPT)
3799  022770  104400                  6$:       ADVANCE                 ;ADVANCE LOOP
3800                                            ;************************ TEST 36 ******************************
3801                                            ;*THIS TEST VERIFIES THAT ODD PARITY WORKS FOR ALL ODD LINES
3802                                            ;* SELECTED AND THAT EVEN PARITY WORKS FOR ALL EVEN LINES SELECTED
3803                                            ;*THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
3804                                            ;*THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
3805                                            ;*THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
3806                                            ;*YOU ARE IN "STAGGERED" MODE.
3807                                            ;*40(8) CHARS ARE USED FOR THIS TEST.
3808                                            ;*ALL SELECTED LINES WILL BE ENABLED
3809                                            ;*AT THE SAME TIME!
3810                                        ;;* TEST 36
3811                                        ;;*****************************************************************
3812  022772  000004                        TST36:  SCOPE
3813  022774  012737  000036  001122                MOV     #36,$TSTNM      ;LOAD THE NUMBER OF THIS TES
3814  023002  012737  004562  001360                MOV     #$EOP,NEXT      ;POINT TO THE END-OF-PASS HANDLER
3815  023010  005737  001370                       TST     MODE            ;IS THIS STAGGERED MODE?
3816  023014  100105                               BPL     6$              ;IF NOT, DON'T DO THIS TEST
3817  023016  104417                               DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3818  023020  013701  001366                       MOV     PAR,R1          ;USE R1 TO BUILD PARAMETERS TO BE LOADED
3819  023024  042701  000200                       BIC     #ODDPAR,R1      ;MAKE SURE ODD PARITY ISN'T SET
3820  023030  052701  000100                       BIS     #PARITY,R1      ;MAKE SURE PARITY IS TURNED ON
3821  023034  012702  000001                       MOV     #1,R2           ;USE R2 AS A LINE POINTER
3822  023040  030237  001364            1$:       BIT     R2,LINE         ;IS THIS A VALID LINE?
3823  023044  001411                               BEQ     3$              ;IF NOT, SKIP TO THE NEXT LINE
3824  023046  032701  000001                       BIT     #BIT0,R1        ;IS THIS LINE AN ODD LINE?
3825  023052  001402                               BEQ     2$              ;IF IT'S EVEN, USE EVEN PARITY
3826  023054  052701  000200                       BIS     #ODDPAR,R1      ;IF IT'S ODD, USE ODD PARITY
3827  023060  010177  156766            2$:       MOV     R1,@DZLPR       ;LOAD THE LINE PARAMETER REGISTER
3828  023064  042701  000200                       BIC     #ODDPAR,R1      ;SET UP THE NEXT PARITY TO EVEN
3829  023070  005201                    3$:       INC     R1              ;POINT TO THE NEXT LINE
3830  023072  106302                               ASLB    R2              ;MOVE THE BIT POINTER IN R2 TO THE NEXT LINE
3831  023074  103361                               BCC     1$              ;IF WE'RE NOT DONE, GO CHECK THE NEXT LINE
3832  023076  005037  001372                       CLR     SAVLIN          ;CLEAR THE LINE NUMBER INDICATOR
3833  023102  005002                               CLR     R2              ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
3834  023104  005003                               CLR     R3              ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
3835  023106  012737  000040  001376                MOV     #40,XMTCNT      ;TRANSMIT A BINARY COUNT PATTERN(00-40)
```

```
3836  023114  012700  001422                        MOV     #TDO,R0          ;POINT TO THE DATA AREA
3837  023120  005020                                CLR     (R0)+            ;CLEAR A DATA WORD
3838  023122  022700  001462                        CMP     #STOP,R0         ;FINISHED ?
3839  023126  001374                                BNE     .-6              ;NO
3840  023130  005000                                CLR     R0               ;CLEAR OFFSET
3841  023132  012777  023410  156736                MOV     #YMTSRV,@DZTIV   ;SET UP THE TRANSMITTER INTERRUPT VECTOR
3842  023140  012777  023232  156724                MOV     #PARESE,@DZRIV   ;SET UP THE RECEIVER INTERRUPT VECTOR
3843  023146  013777  027060  156720                MOV     DZPRT,@DZRIS     ;SET THE INTERRUPT VECTOR STATUS
3844  023154  013777  027060  156716                MOV     DZPRT,@DZTIS     ;SET TRANSMITTER INTERRUPT PRIORITY
3845  023162  052777  040140  156652                BIS     #RIE!TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
3846  023170  113777  001364  156660                MOVB    LINE,@DZTCR      ;ENABLE ALL SELECTED LINES
3847  023176  106427  000000                        MTPS    #0               ;ALLOW INTERRUPTS
3848  023202  032777  000100  156632  5$:           BIT     #RIE,@DZCSR      ;WHEN RX DONE; RIE WILL =0
3849  023210  001407                                BEQ     6$               ;BR IF ALL DONE
3850  023212  005237  023404                        INC     COUNT0
3851  023216  102771                                BVS     5$
3852  023220  105237  023406                        INCB    COUNT1
3853  023224  100366                                BPL     5$
3854  023226  104011                                ERROR   11               ;*RX FAILED TO FINISH (INTERRUPT)
3855  023230  104400                        6$:     ADVANCE                  ;ADVANCE LOOP
```

```
                                                ;RECEIVER SERVICE ROUTINE(PARITY TEST ONLY)

023232  017704  156610          PARESE: MOV     @DZRBUF,R4      ;GET THE CHARACTER
023236  010401                  MOV     R4,R1           ;COPY THE RECEIVED INFORMATION
023240  000301                  SWAB    R1              ;GET THE LINE NUMBER IN THE LOWER BYTE
023242  042701  177770          BIC     #^C<7>,R1       ;ISOLATE THE LINE NUMBER
023246  010137  001372          MOV     R1,SAVLIN       ;FILL LOC. FOR ERROR PRINTOUT
023252  005704                  TST     R4              ;WAS DATA VALID?
023254  100401                  BMI     10$             ;BRANCH IF YES
023256  104023                  ERROR   23              ;ERROR - DATA VALID NOT SET!
023260  006301          10$:    ASL     R1              ;ALIGN IT ON A WORD BOUNDARY
023262  032704  010000          BIT     #PARER,R4       ;PARITY ERROR SHOULD BE SET. IS IT?
023266  001013                  BNE     11$             ;IF SO, GO CHECK CHARACTER
023270  013737  002046  001400  MOV     DZRBUF,REGIST   ;SET UP FOR THE ERROR MESSAGE
023276  010405                  MOV     R4,R5
023300  042705  000377          BIC     #377,R5
023304  156105  001442          BISB    TRO(R1),R5      ;GET THE CORRECT CHARACTER
023310  052705  110000          BIS     #DVALID!PARER,R5 ;BUILD WHAT WAS EXPECTED
023314  104006                  ERROR   6               ;*ERROR- DID NOT GET CORRECT INFORMATION
023316  126104  001442  11$:    CMPB    TRO(R1),R4      ;CHECK THE CHARACTER. IS IT CORRECT?
023322  001407                  BEQ     12$             ;IF SO, GO SET UP NEXT CHARACTER
023324  116105  001442          MOVB    TRO(R1),R5      ;LOAD THE CHARACTER FOR ERROR REPORTING
023330  042705  177400          BIC     #^C<377>,R5     ;CLEAR SIGN EXTEND
023334  042704  177400          BIC     #^C<377>,R4     ;REMOVE THE JUNK FROM R4, THE ACTUAL CHARACTER
023340  104005                  ERROR   5               ;DATA ERROR
023342  005261  001442  12$:    INC     TRO(R1)         ;SET UP THE NEXT CHARACTER
023346  005203                  INC     R3              ;ADD TO THE TOTAL RECEIVED COUNT
023350  005037  023404          CLR     COUNTO          ;RESET COUNTERS TO NEXT
023354  005037  023406          CLR     COUNT1          ;RECIEVER INTERRUPT
023360  032777  040000  156454  BIT     #TIE,@DZCSR     ;ARE TRANSMISSIONS DONE?
023366  001005                  BNE     13$             ;IF NO, GO RECEIVE SOME MORE
023370  020203                  CMP     R2,R3           ;ARE ALL CHARACTERS RECEIVED?
023372  001003                  BNE     13$             ;IF NO, GO RECEIVE SOME MORE
023374  042777  000100  156440  BIC     #RIE,@DZCSR     ;DISABLE RECEIVER INTERRUPTS
023402  000002          13$:    RTI                     ;GO BACK TO RECEIVER WAIT LOOP
023404  000000          COUNTO: 0
023406  000000          COUNT1: 0


                                                ;TRANSMITTER INTERRUPT SERVICE
                                                ;-----------------------------

023410  117701  156430          XMTSRV: MOVB    @HDZCSR,R1      ;GET THE LINE NUMBER. IS THE TRANSMITTER
023414  100411                  BMI     1$              ;REALLY READY? IF SO, GO LOAD THE CHARACTER
023416  013700  001372          MOV     SAVLIN,R0       ;ADJUST LOCATION SAVLIN
023422  042701  177770          BIC     #^C<7>,R1       ;ISOLATE THE LINE NUMBER
023426  010137  001372          MOV     R1,SAVLIN       ;FOR ERROR PRINTOUT
023432  104003                  ERROR   3               ;*TRANSMITTER NOT READY- FALSE INTERRUPT
023434  010037  001372          MOV     R0,SAVLIN       ;RESET SAVLIN TO PREVIOUS VALUE
023440  042701  177770          BIC     #^C<7>,R1       ;ISOLATE THE LINE NUMBER
023444  006301                  ASL     R1              ;MAKE SURE IT REFERENCES A WORD BOUNDARY
023446  116177  001422  156412  MOVB    TDO(R1),@DZTDR  ;LOAD THE CURRENT CHARACTER FOR THIS LINE
023454  005261  001422          INC     TDO(R1)         ;SET UP NEXT CHARACTER FOR THIS LINE
023460  005202                  INC     R2              ;UP THE NUMBER OF TRANSMISSIONS
023462  023761  001376  001422  CMP     XMTCNT,TDO(R1)  ;HAVE WE DONE ALL PATTERNS ON THIS LINE?
```

```
3912  023470  001015                        BNE     4$              ;IF NOT, KEEP ON TRANSMITTING
3913  023472  012700  000001                MOV     #1,R0           ;SET UP A DESELECTION POINTER
3914  023476  006201                        ASR     R1              ;GET THE LINE NUMBER AGAIN
3915  023500  005301                2$:     DEC     R1              ;REDUCE THE COUNT. WAS THIS THE LINE?
3916  023502  100402                        BMI     3$              ;IF SO, GO DISABLE THE ENABLE BIT FOR IT
3917  023504  006300                        ASL     R0              ;MOVE THE POINTER TO THE NEXT LINE
3918  023506  000774                        BR      2$              ;GO CHECK THE NEXT LINE
3919  023510  140077  156342          3$:   BICB    R0,@DZTCR       ;DISABLE THE LINE POINTED TO BY R0
3920  023514  001003                        BNE     4$              ;IF MORE LINES ARE ACTIVE , GO CONTINUE TRANSMIT
3921  023516  042777  040000  156316        BIC     #TIE,@DZCSR     ;IF NOT, DISABLE TRANSMITTER INTERRUPTS
3922  023524  000002                4$:     RTI                     ;RETURN TO THE TIMING LOOP
3923
3924                                        ; RELATIVE TIME BUILDING ROUTINE
3925                                        ; -----------------------------
3926
3927  023526  012737  000004  001222  BUILD: MOV     #4,$TMP2        ;ROTATE 4 BITS BACK INTO $TMP1
3928  023534  006037  001224          1$:   ROR     $TMP3           ;GET THE BITS FROM $TMP3, THE HIGH BYTE
3929  023540  006037  001220                ROR     $TMP1           ;OF THE RELATIVE TIME COUNTER. PUT THEM BACK
3930  023544  005337  001222                DEC     $TMP2           ;INTO $TMP1 USING THE CARRY BIT WITH
3931                                                                ;ROTATE INSTRUCTIONS
3932  023550  001371                        BNE     1$              ;REDUCE COUNT. ALL BITS BACK? IF NOT, GET MORE
3933  023552  000207                        RTS     PC              ;RETURN TO CALLING TEST
3934
```

```
                                                ;RECEIVER SERVICE ROUTINE
                                                ;
                                                ;
023554  105777  156262          RXISR1: TSTB    @DZCSR          ;IS THE RECEIVER REALLY READY?
023560  100401                          BMI     1$              ;IF SO, GO SERVICE IT
023562  104004                          ERROR   4               ;*ERROR- RECEIVER DONE FLAG ISN'T SET
023564  017704  156256          1$:     MOV     @DZRBUF,R4      ;SAVE THE RECEIVER INFORMATION
023570  100401                          BMI     2$              ;IF IT WAS VALID, GO PROCESS IT
023572  104023                          ERROR   23              ;ERROR- DATA VALID WASN'T SET
023574  032704  070000          2$:     BIT     #OVRRUN!FRMERR!PARER,R4 ;ARE ANY ERROR FLAGS SET?
023600  001403                          BEQ     3$              ;IF NOT, GO CONTINUE PROCESSING
023602  013700  002046                  MOV     DZRBUF,R0       ;SET UP FOR ERROR REPORTING
023606  104002                          ERROR   2               ;ERROR- RECEIVER ERROR FLAG SET
023610  010401                  3$:     MOV     R4,R1           ;COPY THE RECEIVER INFORMATION
023612  000301                          SWAB    R1              ;GET THE LINE NUMBER IN THE LOWER BYTE
023614  042701  177770                  BIC     #^C<7>,R1       ;ISOLATE THE LINE NUMBER
023620  006301                          ASL     R1              ;ALIGN IT ON A WORD BOUNDARY
023622  120461  001442                  CMPB    R4,TRO(R1)      ;IS THE CHARACTER WHAT IT SHOULD BE?
023626  001413                          BEQ     4$              ;IF SO,GO CONTINUE PROCESSING
023630  116105  001442                  MOVB    TRO(R1),R5      ;GET WHAT WAS EXPECTED FOR ERROR REPORTING
023634  042705  177400                  BIC     #^C<377>,R5     ;ELIMINATE PROPAGATED SIGN
023640  042704  177400                  BIC     #^C<377>,R4     ;ISOLATE THE ACTUAL CHARACTER
023644  010137  001372                  MOV     R1,SAVLIN       ;GET THE LINE NUMBER OF THE RECEIVER ERROR
023650  006237  001372                  ASR     SAVLIN          ;ALIGN IT CORRECTLY FOR REPORTING
023654  104005                          ERROR   5               ;*DATA ERROR
023656  005261  001442          4$:     INC     TRO(R1)         ;SET UP THE NEXT EXPECTED CHARACTER
023662  005203                          INC     R3              ;INCREMENT THE COUNT OF RECEIVED CHARACTERS
023664  032761  000020  001442        BIT     #20,TRO(R1)     ;HAVE ALL CHARACTERS BEEN RECEIVED?
023672  001402                          BEQ     5$              ;IF NOT, GO RECEIVE SOME MORE
023674  020203                          CMP     R2,R3           ;HAVE WE RECEIVED ALL CHARACTERS?
023676  001401                          BEQ     6$              ;IF SO,GO DETERMINE THE TIMING
023700  000002                  5$:     RTI                     ;GO CONTINUE TIMING AND ALLOW INTERRUPTS
023702  004737  023526          6$:     JSR     PC,BUILD        ;GET THE RELATIVE TIME (SIGNIFICANT BITS)
                                        ;
023706  013700  023766                  MOV     OFFSET,R0       ;GET POINTER
023712  013760  001220  002102          MOV     $TMP1,TMTBL(R0) ; SAVE THIS TEST'S TIME
023720  005737  023766                  TST     OFFSET          ;FIRST TEST?
023724  001414                          BEQ     7$              ;IF NOT, GO CHECK THE TIME
023726  005740                          TST     -(R0)           ;POINT TO THE PREVIOUS  TIME TAKEN
023730  026037  002102  001220          CMP     TMTBL(R0),$TMP1 ;IS THIS TIME WHAT IT SHOULD BE?
023736  101007                          BHI     7$              ;IF SO, GO TO THE NEXT TEST
023740  016005  002102                  MOV     TMTBL(R0),R5    ;PLACE WHAT WAS EXPECTED IN R5
023744  010137  001372                  MOV     R1,SAVLIN       ;GET THE LINE NUMBER OF THE RECEIVER
023750  006237  001372                  ASR     SAVLIN          ;MAKE SURE IT'S THE LINE NUMBER
023754  104021                          ERROR   21              ;TIMING ERROR
023756  042777  000140  156056  7$:     BIC     #RIE!MSENAB,@DZCSR ;DISABLE THE DEVICE
023764  000002                          RTI                     ;RETURN TO THE PROGRAM
023766  000000                  OFFSET: 0
```

```
                        ;DZ11 ECHO/CABLE TEST
                        ;COPYRIGHT 1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

                                ;*STARTING PROCEDURE
                                ;*LOAD PROGRAM
                                ;*LOAD ADDRESS 000210
                                ;*PRESS START
                                ;*PROGRAM WILL TYPE DZ11 ECHO/CABLE TEST
                                ;*PROGRAM WILL TYPE  WHICH TEST- ECHO OR CABLE
                                ;*TYPE IN  E  OR  C   RESPECTIVELY
                                ;*PROGRAM WILL TYPE "VECTOR ADDRESS-"
                                ;*TYPE IN THE ADDRESS OF THE RECEIVER INTERRUPT VECTOR
                                ;*FOR THE DZ11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
                                ;*PROGRAM WILL TYPE "CONTROL REGISTER ADDRESS-"
                                ;*TYPE IN THE ADDRESS OF THE SYSTEM CONTROL REGISTER
                                ;*FOR THE DZ11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
                                ;*PROGRAM WILL TYPE "LINE NUMBER-"
                                ;*TYPE IN THE LINE NUMBER TO BE TESTED (IN OCTAL)
                                ;*,FOLLOWED BY <CARRIAGE RETURN>
                                ;*PROGRAM WILL TYPE "BAUD RATE-"
                                ;*TYPE IN THE BAUD RATE OF THE DZ11 TERMINAL
                                ;*,FOLLOWED BY <CARRIAGE RETURN>
                                        ;*THE FOLLOWING BAUD RATES ARE ACCEPTED IN DECIMAL
                                ;*              50
                                ;*              75
                                ;*              110
                                ;*              135     (ROUNDED OFF   134.5)
                                ;*              150
                                ;*              300
                                ;*              600
                                ;*              1200
                                ;*              1800
                                ;*              2000
                                ;*              2400
                                ;*              3600
                                ;*              4800
                                ;*              7200
                                ;*              9600
                                ;*ALL OTHERS ARE REJECTED

                                ;*PROGRAM WILL TYPE "ECHO"  OR  "CABLE TEST" TO INDICATE THAT TESTING HAS STARTE


                                ;PROGRAM INITIALIZATION
                                ;LOCK OUT INTERRUPTS
                                ;SET UP PROCESSOR STACK
                                ;SET UP POWER FAIL VECTOR
                                ;CLEAR PROGRAM FLAGS AND COUNTS

023770  012706  001120         XSTART: MOV     #STACK,SP       ;SET UP PROCESSOR STACK
023774  106427  000340                 MTPS    #PR7            ;LOCK OUT INTERRUPTS
024000  012737  023770  001126         MOV     #XSTART,$LPADR  ;SET UP IN CASE OF POWER FAIL
024006  005037  026164                 CLR     STFLG           ;CLEAR TEST START FLAG
024012  005037  001242                 CLR     $PASS           ;CLEAR PASS COUNT
024016  005037  001132                 CLR     $ERTTL          ;CLEAR ERROR COUNT
024022  105037  001123                 CLRB    $ERFLG          ;CLEAR ERROR FLAG
```

```
4038  024026  005037  026170                    CLR     LAST            ;CLEAR LAST ERROR PC
4039  024032  032777  000001  155120  VEC1:     BIT     #SW00,@SWR      ;IF SW00=1, GET NEW VECTOR
4040  024040  001465                            BEQ     OTHER           ;AND CSR
4041  024042  012701  000300          VEC2:     MOV     #300,R1
4042  024046  012702  000302                    MOV     #302,R2
4043  024052  010221                  1$:       MOV     R2,(R1)+        ;RESTORE TRAPCATCHER
4044  024054  005022                            CLR     (R2)+           ;IN FLOATING VECTOR AREA
4045  024056  022122                            CMP     (R1)+,(R2)+     ;UPDATE THE POINTERS
4046  024060  020127  001000                    CMP     R1,#1000
4047  024064  001372                            BNE     1$
4048  024066  104403                            INSTR                   ;INPUT ADDRESS OF DEVICE VECTOR
4049  024070  026216                            MVECTOR                 ;MESSAGE "VECTOR ADDRESS-"
4050  024072  104405                            PARAM                   ;CONVERT STRING TO OCTAL
4051  024074  000300                            300                     ;LOW LIMIT
4052  024076  000770                            770                     ;HIGH  LIMIT
4053  024100  002072                            DZRIV                   ;LOCATIONS TO BE FILLED
4054  024102     003                  .BYTE     3                       ;LSB MASK
4055  024103     004                  .BYTE     4                       ;NUMBER OF LOCATIONS
4056  024104  104403                            INSTR                   ;INPUT ADDRESS OF DEVICE CSR
4057  024106  026240                            MREGAD                  ;MESSAGE "CONTROL REGISTER ADDRESS-"
4058  024110  104405                            PARAM                   ;CONVERT STRING TO OCTAL
4059  024112  160000                            160000                  ;LOW LIMIT
4060  024114  163700                            163700                  ;HIGH LIMIT
4061  024116  002042                            DZCSR                   ;LOCATIONS TO BE FILLED
4062  024120     007                  .BYTE     7                       ;LSB MASK
4063  024121     001                  .BYTE     1                       ;NUMBER OF LOCATIONS
4064  024122  013737  002042  002046            MOV     DZCSR,DZRBUF    ;BEGIN BUILDING DEVICE ADDRESSES
4065  024130  062737  000002  002046            ADD     #2,DZRBUF       ;FORM THE READ BUFFER ADDRESS
4066  024136  013737  002046  002052            MOV     DZRBUF,DZLPR    ;REMEMBER THAT THIS IS ALSO LINE PARAMETER REG.
4067  024144  013737  002046  002056            MOV     DZRBUF,DZTCR    ;BEGIN BUILDING TRANSMITTER CONTROL REGISTER
4068  024152  062737  000002  002056            ADD     #2,DZTCR        ;FORM THE TRANSMITTER CONTROL REGISTER POINTER
4069  024160  013737  002056  002060            MOV     DZTCR,HDZTCR
4070  024166  005237  002060                    INC     HDZTCR
4071  024172  013737  002056  002066            MOV     DZTCR,DZTDR     ;BEGIN FORMING TRANSMITTER DATA REGISTER
4072  024200  062737  000002  002066            ADD     #2,DZTDR        ;FORM THE TRANSMITTER DATA REGISTER
4073  024206  013737  002066  002062            MOV     DZTDR,DZMSR
4074  024214  032777  000002  154736  OTHER:    BIT     #SW01,@SWR      ;RESELECT OF TEST?
4075  024222  001427                            BEQ     XBEGIN          ;IF NOT, SKIP ASKING WHICH ONE
4076  024224  104403                            INSTR                   ;INPUT WHICH TEST YOU ARE RUNNING
4077  024226  026424                            MWHICH                  ;ECHO OR CABLE
4078  024230  104416                            PAWCH                   ;SET FLAG
4079  024232  026162                            WCHFLG                  ;THIS FLAG
4080  024234  104403                  BAUD:     INSTR                   ;INPUT BAUD RATE
4081  024236  026346                            MSPEED                  ;MESSAGE "BAUD RATE-"
4082  024240  104415                            PARMD                   ;CONVERT DECIMAL STRING TO OCTAL
4083  024242  000062                            50.                     ;LOW LIMIT
4084  024244  022600                            9600.                   ;HIGH LIMIT
4085  024246  026200                            LINESP                  ;LOCATION TO BE FILLED
4086  024250     000                  .BYTE     0                       ;LSB MASK
4087  024251     001                  .BYTE     1                       ;NUMBER OF LOCATIONS
4088  024252  104413                  LINEX:    DEVICE.CLR              ;CLEAR DEVICE
4089  024254  005037  026164                    CLR     STFLG           ;CLEAR PROGRAM START FLAG
4090  024260  104403                            INSTR                   ;INPUT LINE NUMBER
4091  024262  026336                            MLINE                   ;MESSAGE "LINE NUMBER-"
4092  024264  104405                            PARAM                   ;CONVERT  STRING TO OCTAL
4093  024266  000000                            0                       ;LOW LIMIT
```

```
4094  024270  000007                          7                       ;HIGH LIMIT
4095  024272  001372                          SAVLIN                  ;LOCATION TO BE FILLED
4096  024274     000                .BYTE      0                       ;LSB MASK
4097  024275     001                .BYTE      1                       ;NUMBER OF LOCATIONS
4098  024276  004537  025766                   JSR       R5,SET
4099
4100  024302  106427  000340      XBEGIN:     MTPS      #PR7          ;LOCK OUT INTERRUPTS
4101  024306  012706  001120                  MOV       #STACK,SP     ;SET UP PROCESSOR STACK
4102  024312  005037  026166                  CLR       LOCKUP        ;CLEAR TIMEOUT
4103  024316  005737  026162                  TST       WCHFLG        ;ECHO OR CABLE TEST ?
4104  024322  001413                          BEQ       2$            ;ECHO
4105  024324  012737  025040  001126          MOV       #TEST2,$LPADR ;CABLE TEST
4106  024332  005737  026164                  TST       STFLG         ;ARE YOU LOOPING ?
4107  024336  001017                          BNE       1$            ;YES
4108  024340  005137  026164                  COM       STFLG         ;NO
4109  024344  104402  026517                  TYPE      ,MCABLE       ;TYPE CABLE TEST
4110  024350  000412                          BR        1$
4111  024352  012737  024402  001126  2$:     MOV       #TEST1,$LPADR ;SET UP ECHO TEST
4112  024360  005737  026164                  TST       STFLG         ;ARE YOU LOOPING ?
4113  024364  001004                          BNE       1$            ;YES
4114  024366  005137  026164                  COM       STFLG         ;NO
4115  024372  104402  026472                  TYPE      ,MTERM        ;TYPE ECHO TEST
4116  024376  000177  154524      1$:         JMP       @$LPADR       ;START TESTING
4117                                          ;THIS TEST WILL ACCEPT 1 CHARACTER AT A TIME
4118                                          ;(IN INTERRUPT MODE) AND TRANSMIT THAT SAME CHARACTER,
4119                                          ;ONE LINE AT A TIME, ANY LINE 0 THRU 7 (OCTAL)
4120
4121  024402  104413              TEST1:      DEVICE.CLR              ;CLEAR DZ11
4122  024404  012737  000001  001122          MOV       #1,$TSTNM
4123  024412  013777  026206  155436          MOV       NUMTCR,@DZTCR ;SET TCR BIT
4124  024420  013737  026204  001366          MOV       NUMLIN,PAR    ;SET PARAMETERS
4125  024426  053737  026202  001366          BIS       SPEED,PAR     ;SET BAUD RATE
4126  024434  013777  001366  155410          MOV       PAR,@DZLPR    ;LOAD PARAM.
4127  024442  012777  000040  155372          MOV       #MSENAB,@DZCSR ;SET SCANN ENABLE
4128  024450  005004                          CLR       R4
4129  024452  012705  026534                  MOV       #MQUICK,R5    ;SET MESSAGE BUFFER
4130  024456  005777  155360      3$:         TST       @DZCSR        ;TRDY?
4131  024462  100404                          BMI       2$            ;BR IF YES
4132  024464  104414                          DELAY                   ;WAIT
4133  024466  005304                          DEC       R4            ;
4134  024470  001372                          BNE       3$            ;
4135  024472  104003                          ERROR     3             ;NO TRDY SET! WHY?
4136  024474  005004      2$:                 CLR       R4            ;RESET COUNTER TO 0
4137  024476  112577  155364                  MOVB      (R5)+,@DZTDR  ;LOAD CHAR
4138  024502  001365                          BNE       3$            ;
4139  024504  004737  007360                  JSR       PC,SERV.G     ;<↑G>?
4140  024510  122777  000377  154442          CMPB      #377,@SWR     ;RE-DO QUICK BROWN?
4141  024516  001731                          BEQ       TEST1         ;BR IF REPEAT PATTERN
4142  024520  104413                          DEVICE.CLR
4143  024522  106427  000340                  MTPS      #PR7          ;LOCK OUT INTERRUPTS
4144  024526  012737  025476  001360          MOV       #XEOP,NEXT
4145  024534  104413                          DEVICE.CLR
4146  024536  013737  026204  001366          MOV       NUMLIN,PAR    ;SELECT LINE # & SET INTERRUPT ENABLE
4147  024544  053737  026202  001366          BIS       SPEED,PAR     ;SET LINE SPEED AND
4148                                                                  ;CHARACTER LENGTH (TRANS. & REC.)
4149  024552  052737  010000  001366          BIS       #RCVON,PAR    ;MAKE SURE RECEIVER IS TURNED ON
```

```
4150  024560  013727  001366  155264          MOV     PAR,@DZLPR      ;LOAD THE LINE PARAMETER REGISTER
4151  024566  012777  024642  155276          MOV     #INTSVC,@DZRIV  ;SET UP INTERRUPT SERVICE
4152  024574  013777  026210  155272          MOV     PRIO,@DZRIS     ;AND LEVEL
4153  024602  106437  027062                  MTPS    @#LESS1         ;ALLOW INTERRUPTS
4154  024606  012777  000140  155226          MOV     #RIE!MSENAB,@DZCSR ;SET RECEIVER INTERRUPT ENABLE
4155  024614  104402  026364                  TYPE    ,MCHAR          ;TYPE "ANY CHARACTER"
4156  024620  105777  154340          1$:     TSTB    @$TKS           ;IF SOMBODY HITS A KEY- GET NEW LINE #
4157  024624  100375                          BPL     1$              ;LOOP HERE
4158  024626  005777  154334                  TST     @$TKB           ;CLEAR CHAR
4159  024632  004737  007360                  JSR     PC,SERV.G       ;MAKE SURE IT WASN'T <^G>
4160  024636  000137  024252                  JMP     LINEX           ;
4161
4162
4163                                          ;THE FOLLOWING IS THE RECEIVER INTERRUPT SVC ROUTINE
4164  024642  105777  155174          INTSVC: TSTB    @DZCSR          ;TEST REC. FLAG
4165  024646  100401                          BMI     .+4
4166  024650  104004                          ERROR   4               ;ERROR - INTERRUPT NOT CAUSED BY FLAG
4167  024652  017737  155170  026212          MOV     @DZRBUF,RECDAT
4168  024660  100401                          BMI     .+4
4169  024662  104023                          ERROR   23              ;NON- VALID CHARACTER
4170  024664  032737  020000  026212          BIT     #BIT13,RECDAT   ;CHECK FOR FRAMING ERROR
4171  024672  001401                          BEQ     .+4             ;BR IF NO ERROR
4172  024674  104025                          ERROR   25              ;EITHER SOMBODY HIT THE
4173                                                                  ;"BREAK KEY" OR YOU HAVE AN ERROR!
4174  024676  113737  026212  026214          MOVB    RECDAT,TBUF     ;MOVE CHARACTER TO OUTPUT AREA
4175  024704  113737  026212  010620          MOVB    RECDAT,INBUF    ;MOVE CHARACTER TO CHECK FOR ^C
4176  024712  042737  177600  010620          BIC     #^C<177>,INBUF  ;STRIP JUNK PLUS PARITY
4177  024720  042737  174377  026212          BIC     #174377,RECDAT  ;SAVE ONLY LINE  NUMBER
4178  024726  000337  026212                  SWAB    RECDAT
4179  024732  023737  001372  026212          CMP     SAVLIN,RECDAT   ;DOES THE LINE # COMPARE?
4180  024740  001401                          BEQ     .+4
4181  024742  104015                          ERROR   15              ;*WRONG LINE NUMBER
4182  024744  012777  000040  155070          MOV     #MSENAB,@DZCSR  ;START THE TRANSMITTERS SCANNER
4183  024752  123727  010620  000003          CMPB    INBUF,#3                ;IS IT A ^C ?
4184  024760  001004                          BNE     1$              ;NO
4185  024762  104413                          DEVICE.CLR
4186  024764  012716  025476                  MOV     #XEOP,(SP)      ;CRUNCH STACK
4187  024770  000002                          RTI
4188  024772  005003                  1$:     CLR     R3              ;INITIALIZE DELAY
4189  024774  013777  026206  155054          MOV     NUMTCR,@DZTCR   ;ENABLE THE LINE
4190  025002  005777  155034          10$:    TST     @DZCSR          ;TRANSMITTER READY?
4191  025006  100403                          BMI     2$              ;IF YES BRANCH
4192  025010  005203                          INC     R3              ;INCREMENT DELAY
4193  025012  001373                          BNE     10$             ;DELAY DONE?
4194  025014  104003                          ERROR   3               ;TRANSMIT READY NOT SET!
4195  025016  113777  026214  155042  2$:     MOVB    TBUF,@DZTDR     ;TRANSMIT THE CHARACTER
4196  025024  012777  000140  155010          MOV     #RIE!MSENAB,@DZCSR      ;RESTART THE RECEIVER
4197  025032  005077  155020                  CLR     @DZTCR          ;CLEAR TCR BIT
4198  025036  000002                          RTI
4199
4200
4201                                          ;THIS TEST TRANSMITS A BINARY COUNT PATTERN
4202                                          ;VIA INTERRUPT MODE TO THE RECEIVER
4203                                          ;...THE LINE UNDER TEST MUST BE TERMINATED WITH THE TEST CONNECTOR
4204  025040  106427  000340          TEST2:  MTPS    #PR7            ;DISABLE INTERRUPTS
4205  025044  012737  000002  001122          MOV     #2,$TSTNM
```

```
4206  025052  012737  025476  001360          MOV     #XEOP,NEXT
4207  025060  104413                          DEVICE.CLR              ;
4208                                          ;*TEST TO VERIFY THAT SETTING DTR FOR A GIVEN LINE
4209                                          ;*WILL BRING UP "CO" AND "RING" FOR THE SAME LINE
4210                                          ;*THE DIST PNL MUST HAVE JUMPER FROM DTR TO RQST TO SEND
4211                                          ;*IN ORDER FOR THIS TEST TO WORK!
4212  025062  012737  025070  001362          MOV     #1$,LOCK        ;LOOP
4213  025070  113777  026206  154762  1$:     MOVB    NUMTCR,@HDZTCR  ;SET DTR
4214  025076  005005                          CLR     R5              ;
4215  025100  153705  026206                  BISB    NUMTCR,R5       ;BUILD EXPECTED
4216  025104  000305                          SWAB    R5              ;PUT IN HIGH BYTE
4217  025106  153705  026206                  BISB    NUMTCR,R5       ;
4218  025112  104414                          DELAY                   ;WAIT FOR CABLE DELAY
4219  025114  017704  154742                  MOV     @DZMSR,R4       ;READY MODEM BITS
4220  025120  020504                          CMP     R5,R4           ;ARE THEY OK?
4221  025122  001401                          BEQ     2$              ;BR IF YES
4222  025124  104022                          ERROR   22              ;IS THE TEST CONNECTOR ON?
4223                                                                  ;HAS RIGHT LINE BEEN SELECTED?
4224                                                                  ;IF SO- YOU HAVE A PROBLEM!
4225                                                                  ;MODEM BITS NOT RIGHT
4226  025126  104401                  2$:     SCOP1                   ;LOOP
4227  025130  104413                  3$:     DEVICE.CLR              ;INIT DZ11
4228  025132  013737  026202  001366          MOV     SPEED,PAR       ;SET LINE SPEED
4229  025140  053737  026204  001366          BIS     NUMLIN,PAR      ;SELECT LINE # & REC.  INTERRUPT ENABLE
4230  025146  052737  010000  001366          BIS     #RCVON,PAR      ;ENABLE THE RECEIVER FOR THIS LINE
4231  025154  052777  040140  154660          BIS     #TIE!RIE!MSENAB,@DZCSR ;SET TRANSMITTER INTERRUPT ENABLE
4232  025162  012777  025276  154702          MOV     #INTREC,@DZRIV  ;SET UP INTR SERVICE
4233  025170  013777  026210  154676          MOV     PRIO,@DZRIS     ;SET UP LEVEL
4234  025176  012777  025456  154672          MOV     #INTRAN,@DZTIV  ;SET UP INTR SERVICE
4235  025204  013777  026210  154666          MOV     PRIO,@DZTIS     ;SET UP LEVEL
4236  025212  005001                          CLR     R1              ;RX DATA POINTER- SET TO 0
4237  025214  005002                          CLR     R2              ;TX DATA POINTER- SET TO 0
4238  025216  013777  001366  154626          MOV     PAR,@DZLPR      ;SET THE PARAMETERS AND TURN ON RECEIVER
4239  025224  106437  027062                  MTPS    @#LESS1         ;ALLOW INTERRUPTS
4240  025230  013777  026206  154620          MOV     NUMTCR,@DZTCR   ;SET UP TCR BIT
4241
4242                                          ;YOU RETURN HERE AFTER EVERY RECEIVER INTERRUPT
4243  025236  105777  153722          SPIN:   TSTB    @#TKS           ;IF SOMEBODY HITS A KEY- GET A NEW LINE #
4244  025242  100006                          BPL     1$              ;BR IF NO KEY HIT
4245  025244  005777  153716                  TST     @#TKB           ;CLEAR CHAR
4246  025250  004737  007360                  JSR     PC,SERV.G       ;MAKE SURE IT WASN'T <↑G>
4247  025254  000137  024252                  JMP     LINEX           ;SW02=1
4248  025260  005237  026166          1$:     INC     LOCKUP          ;INC TIMEOUT FLAG
4249  025264  001364                          BNE     SPIN            ;IF NOT 0  RETURN SPINNING
4250  025266  104011                          ERROR   11              ;*RECEIVER FAILED TO INTERRUPT  CHECK CABLE/TERMINATOR
4251  025270  104413                  QUITS:  DEVICE.CLR
4252  025272  000137  025476                  JMP     XEOP            ;CALL FOR END OF PASS
4253  025276  005037  026166          INTREC: CLR     LOCKUP          ;CLEAR TIMEOUT FLAG
4254  025302  105777  154534                  TSTB    @DZCSR          ;TEST REC DONE
4255  025306  100401                          BMI     .+4             ;YES
4256  025310  104004                          ERROR   4               ;*FALSE INTERRUPT
4257  025312  017737  154530  026212          MOV     @DZRBUF,RECDAT  ;SAVE WORD
4258  025320  100401                          BMI     .+4
4259  025322  104023                          ERROR   23              ;*NON VALID CHARACTER
4260  025324  032737  040000  026212          BIT     #BIT14,RECDAT   ;DATA OVERRUN ?
4261  025332  001401                          BEQ     .+4             ;NO
```

```
4262  025334  104024                          ERROR   24                      ;*YES
4263  025336  032737  020000  026212          BIT     #BIT13,RECDAT           ;FRAMING ERROR ?
4264  025344  001401                          BEQ     .+4                     ;NO
4265  025346  104025                          ERROR   25                      ;*YES
4266  025350  032737  010000  026212          BIT     #BIT12,RECDAT           ;PARITY ERROR ?
4267  025356  001401                          BEQ     .+4                     ;NO
4268  025360  104026                          ERROR   26                      ;*YES
4269  025362  110105                          MOVB    R1,R5                   ;SET EXPECTED
4270  025364  042705  177400                  BIC     #^C<377>,R5             ;CLEAR HIGH BYTE
4271  025370  113704  026212                  MOVB    RECDAT,R4               ;GET FOUND
4272  025374  042704  177400                  BIC     #^C<377>,R4             ;CLEAR HIGH BYTE
4273  025400  020504                          CMP     R5,R4   ;OK?
4274  025402  001401                          BEQ     .+4
4275  025404  104005                          ERROR   5                       ;DATA ERROR
4276  025406  042737  174377  026212          BIC     #174377,RECDAT          ;SAVE ONLY LINE NUMBER
4277  025414  000337  026212                  SWAB    RECDAT
4278  025420  023737  001372  026212          CMP     SAVLIN,RECDAT           ;DOES THE LINE # COMPARE ?
4279  025426  001401                          BEQ     .+4                     ;YES
4280  025430  104015                          ERROR   15                      ;*WRONG LINE #
4281  025432  120127  000377                  CMPB    R1,#377                 ;LAST CHARACTER ?
4282  025436  001003                          BNE     1$                      ;NO
4283  025440  012716  025270                  MOV     #QUITS,(SP)             ;CRUNCH STACK
4284  025444  000403                          BR      2$
4285  025446  105201                  1$:     INCB    R1                      ;UPDATE EXPECTED DATA
4286  025450  012716  025236                  MOV     #SPIN,(SP)              ;CRUNCH STACK
4287  025454  000002                  2$:     RTI
4288
4289  025456  005777  154360          INTRAN: TST     @DZCSR  ;TEST TRANSMIT FLAG
4290  025462  100401                          BMI     .+4
4291  025464  104003                          ERROR   3                       ;*FALSE INTERRUPT
4292  025466  110277  154374                  MOVB    R2,@DZTDR               ;TRANSMIT A CHARACTER
4293  025472  105202                          INCB    R2                      ;UPDATE TX DATA
4294  025474  000002                          RTI     ;RETURN
```

```
                                        ;END OF PASS
                                        ;RESTART TEST

025476  104402          XEOP:   TYPE                    ;TYPE NAME OF TEST
025500  026274                  MPASS
025502  005037  026170          CLR     LAST            ;CLEAR LAST ERROR PC
025506  105037  001123          CLRB    $ERFLG          ;CLEAR ERROR FLAG
025512  000137  024302  RSTRT:  JMP     XBEGIN

                                        ;CONVERT DECIMAL ASCII STRING TO OCTAL
025516  011605          .PARMD: MOV     (SP),R5
025520  012537  025702          MOV     (R5)+,6$
025524  012537  025704          MOV     (R5)+,7$
025530  012537  025706          MOV     (R5)+,8$
025534  112537  025710          MOVB    (R5)+,9$
025540  112537  025711          MOVB    (R5)+,10$
025544  010516                  MOV     R5,(SP)
025546  005005          2$:     CLR     R5
025550  012704  010620          MOV     #INBUF,R4
025554  122714  000015          CMPB    #15,(R4)
025560  001424                  BEQ     3$
025562  121427  000060  1$:     CMPB    (R4),#'0
025566  002421                  BLT     3$
025570  121427  000071          CMPB    (R4),#'9
025574  003016                  BGT     3$
025576  142714  000060          BICB    #'0,(R4)
025602  005002                  CLR     R2
025604  157402                  BISB    (R4)+,R2
025606  060205                  ADD     R2,R5
025610  122714  000015          CMPB    #15,(R4)
025614  001410                  BEQ     4$
025616  006305                  ASL     R5              ;X2
025620  010502                  MOV     R5,R2           ;SAVE X2
025622  006305                  ASL     R5              ;X4
025624  006305                  ASL     R5              ;X8
025626  060205                  ADD     R2,R5           ;TIMES 10
025630  000754                  BR      1$
025632  104404          3$:     INSTER
025634  000744                  BR      2$

                                        ;TEST TO SEE IF NUMBER IS WITHIN LIMITS

025636  020537  025704  4$:     CMP     R5,7$
025642  101373                  BHI     3$
025644  020537  025702          CMP     R5,6$
025650  103770                  BLO     3$
025652  133705  025710          BITB    9$,R5
025656  001365                  BNE     3$

                                        ;STORE NUMBER AT SPECIFIED ADDRESS

025660  013704  025706          MOV     8$,R4
025664  010524  5$:     MOV     R5,(R4)+
025666  062705  000002          ADD     #2,R5
025672  105337  025711          DECB    10$
```

```
                        001372                  BNE     5$
                        000002                  RTI
                        000000          6$:     0
                        000000          7$:     0
                        000000          8$:     0
                           000          9$:     .BYTE 0
                           000          10$:    .BYTE 0


                                        ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
                                        ;BUFFER TO THE CHARACTERS "E" AND "C".
                                        ;IF THE CHARACTER IS "E" CLEAR THE FLAG
                                        ;IF THE CHARACTER IS "C" SET THE FLAG

                        017605  000000  .PAWCH: MOV     @(SP),R5
                        142737  000040  010620  BICB    #40,INBUF       ;SET FOR LOWER CASE INPUT
                        122737  000105  010620  CMPB    #'E,INBUF       ;IS IT "E" ?
                        001002                  BNE     1$
                        105015                  CLRB    (R5)            ;000
                        000406                  BR      2$
                        122737  000103  010620  1$:     CMPB    #'C,INBUF       ;IS IT "C" ?
                        001005                  BNE     3$
                        112715  177777          MOVB    #-1,(R5)        ;3177
                        062716  000002          2$:     ADD     #2,(SP)
                        000002                  RTI
                        104404                  3$:     INSTER                  ;RETRY
                        000752                  BR      .PAWCH


                                        ;THIS ROUTINE CONVERTS LINE SPEED (LINESP) AND
                                        ;LINE NUMBER (SAVLIN) FOR DZLPR, DZTCR AND DZCSR
                                        ;REGISTER USAGE.

                        013737  001372  026204  SET:    MOV     SAVLIN,NUMLIN   ;SAVE SAVLIN
                        013700  001372          XTCR0:  MOV     SAVLIN,R0       ;COPY THE LINE NUMBER FOR LOOP CONTROL
                        005037  026206                  CLR     NUMTCR          ;SET A DEFAULT OF LINE 0 OR NO LINES
                        012702  000001                  MOV     #1,R2           ;SET A BIT POINTER TO THE FIRST LINE
                        005300                  XTCR1:  DEC     R0              ;REDUCE THE INDICATOR.IS IT MINUS \ T?
                        100402                          BMI     SET1            ;IF SO, R2 POINTS TO THE RIGHT LINE
                        006302                          ASL     R2              ;IF NOT, MOVE THE POINTER TO THE NEXT LINE
                        000774                          BR      XTCR1           ;GO SEE IF THIS LINE IS THE ONE
                        012701  026062          SET1:   MOV     #TABLE2,R1
                        010237  026206                  MOV     R2,NUMTCR       ;COPY THE CORRECT BIT POINTER
                        022137  026200          1$:     CMP     (R1)+,LINESP
                        001407                          BEQ     2$
                        005721                          TST     (R1)+           ;IS IT THE END OF TABLE?
                        001373                          BNE     1$              ;NO
                        104402  026310                  TYPE    ,MINVAL         ;INVALID BAUD RATE,BEGIN AGAIN
                        012705  024234                  MOV     #BAUD,R5        ;JUMP TO BAUD THRU R5
                        000402                          BR      3$
                        011137  026202          2$:     MOV     (R1),SPEED      ;SET UP BAUD RATE
                        000205          3$:     RTS     R5
```

```
                                        ;THE FOLLOWING IS A TABLE OF LEGAL BAUD RATES (8 BITS/CHAR)
026062  000062          TABLE2: .WORD   50.             ;50 BAUD
026064  010070                  .WORD   10070           ;
026066  000113                  .WORD   75.             ;75 BAUD
026070  010470                  .WORD   10470           ;
026072  000156                  .WORD   110.            ;110 BAUD
026074  011070                  .WORD   11070           ;TWO STOP BITS
026076  000207                  .WORD   135.            ;134.5 BAUD
026100  011470                  .WORD   11470           ;TWO STOP BITS
026102  000226                  .WORD   150.            ;150 BAUD
026104  012070                  .WORD   12070           ;TWO STOP BITS
026106  000454                  .WORD   300.            ;300 BAUD
026110  012430                  .WORD   12430           ;ONE STOP BIT
026112  001130                  .WORD   600.            ;600 BAUD
026114  013030                  .WORD   13030           ;ONE STOP BIT
026116  002260                  .WORD   1200.           ;1200 BAUD
026120  013430                  .WORD   13430           ;ONE STOP BIT
026122  003410                  .WORD   1800.           ;1800 BAUD
026124  014030                  .WORD   14030           ;ONE STOP BIT
026126  003720                  .WORD   2000.           ;2000 BAUD
026130  014430                  .WORD   14430           ;ONE STOP BIT
026132  004540                  .WORD   2400.           ;2400 BAUD
026134  015030                  .WORD   15030           ;ONE STOP BIT
026136  007020                  .WORD   3600.           ;3600 BAUD
026140  015430                  .WORD   15430           ;ONE STOP BIT
026142  011300                  .WORD   4800.           ;4800 BAUD
026144  016030                  .WORD   16030           ;ONE STOP BIT
026146  016040                  .WORD   7200.           ;7200 BAUD
026150  016430                  .WORD   16430           ;ONE STOP BIT
026152  022600                  .WORD   9600.           ;9600 BAUD
026154  017070                  .WORD   17070           ;
026156  177777  000000          .WORD   -1,0            ;TABLE TERMINATOR

026162  000000          WCHFLG:0                        ;ECHO OR CABLE FLAG
026164  000000          STFLG:  0                       ;PROGRAM START FLAG
026166  000000          LOCKUP: 0                       ;TIMEOUT FLAG
026170  000000          LAST:   0                       ;LAST ERROR PC
026172  000000          TDATA:  0
026174  000000          RDATA:  0
026176  000000          BYTCNT: 0
026200  000156          LINESP: 110.                    ;DEFAULT BAUD RATE
026202  006307          SPEED:  6307                    ;DEFAULT 110 BAUD, 8 BITS/CHAR,
                                                        ;FDX, 2 STOP BITS
026204  000100          NUMLIN: 100                     ;DEFAULT VALUE, REC. INTERRUPT ENABLED

026206  000001          NUMTCR: 1                       ;DEFAULT VALUE,TCR BIT 0
026210  000240          PRIO:   240                     ;DEFAULT DEVICE PRIORITY 5
026212  000000          RECDAT: 0
026214  000000          TBUF:   0
026216  053200  041505  047524  MVECTO: .ASCIZ  <200>/VECTOR ADDRESS- /
026240  041600  047117  051124  MREGAD: .ASCIZ  <200>/CONTROL REGISTER ADDRESS- /
026274  050200  051501  020123  MPASS:  .ASCIZ  <200>/PASS DONE./
026310  044600  053116  046101  MINVAL: .ASCIZ  <200>/INVALID BAUD RATE - /
026336  046200  047111  035105  MLINE:  .ASCIZ  <200>/LINE: /
026346  041200  052501  020104  MSPEED: .ASCIZ  <200>/BAUD RATE - /
```

```
026764  052200  050131  020105  MCHAR:  .ASCIZ  <200>/TYPE A CHAR. ON DZ11 TERMINAL /
026424  053600  044510  044103  MWHICH: .ASCIZ  <200>/WHICH TEST ? ECHO OR CABLE (E OR C) /
026472  052200  051105  044515  MTERM:  .ASCIZ  <200>/TERMINAL ECHO TEST /
026517     200  040503  046102  MCABLE: .ASCIZ  <200>/CABLE TEST /
026534  006777  177777  177412  MQUICK: .ASCII  <377><15><377><377><12><377><377>
026543     124  042510  050440          .ASCII  /THE Q"ICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 0123456789/
026640  006777  177777  177412          .ASCII  <377><.5><377><377><12><377><377><377><0>
        026652                  .EVEN
                                ;:**********************************************************
                                ;UTILITIES
                                ;:**********************************************************

                                ;THIS UTILITY CALCULATES PRIORITY LEVEL,SETS UP CSR'S,SETS UP VECTORS.
026652  006337  027060  DZLEV:  ASL     DZPRT           ;BUILD PRIORITY IN THIS LOCATION
026656  006337  027060          ASL     DZPRT           ;USING ARITHMETIC SHIFTS, ROTATE
026662  006337  027060          ASL     DZPRT           ;       THE PRIORITY LEVEL PAST
026666  006337  027060          ASL     DZPRT           ;       THE BIT POSITIONS CORRE-
026672  006337  027060          ASL     DZPRT           ;       SPONDING TO THE CONDITION CODES
026676  013737  027060  027062  MOV     DZPRT,LESS1     ;MOVE THIS TO LESS1
026704  162737  000001  027062  SUB     #1,LESS1        ;CREATE THE NEXT LOWEST PRIORITY
026712  042737  000037  027062  BIC     #37,LESS1       ;INSURE THAT THE  I,N,Z,V,C BITS ARE CLEAR
026720  013700  002072          MOV     DZRIV,R0        ;PLACE THE BASE VECTOR ADDRESS IN R0
026724  062700  000002          ADD     #2,R0           ;CALCULATE THE RECEIVER INTERRUPT STATUS ADDR.
026730  010037  002074          MOV     R0,DZRIS        ;STORE IT HERE
026734  062700  000002          ADD     #2,R0           ;CALCULATE THE TRANSMITTER INTERRUPT VECTOR
026740  010037  002076          MOV     R0,DZTIV        ;STORE IT HERE
026744  062700  000002          ADD     #2,R0           ;CALCULATE THE TRANSMITTER VECTOR STATUS ADDRESS
026750  010037  002100          MOV     R0,DZTIS        ;STORE IT HERE

                                ;THIS SEGMENT SETS UP POINTERS FOR THE GIVEN DZ11. $BASE IS THE BASE ADDRESS
                                ;OF THE DEVICE
026754  013700  001310          MOV     $BASE,R0        ;COPY THE ADDRESS BEING LOADED
026760  010037  002042          MOV     R0,DZCSR        ;XXX0
026764  005200                  INC     R0
026766  010037  002044          MOV     R0,HDZCSR       ;XXX1
026772  005200                  INC     R0
026774  010037  002046          MOV     R0,DZRBUF       ;XXX2
027000  010037  002052          MOV     R0,DZLPR        ;XXX2
027004  005200                  INC     R0
027006  010037  002050          MOV     R0,HDZRBUF      ;XXX3
027012  010037  002054          MOV     R0,HDZLPR       ;XXX3
027016  005200                  INC     R0
027020  010037  002056          MOV     R0,DZTCR        ;XXX4
027024  005200                  INC     R0
027026  010037  002060          MOV     R0,HDZTCR       ;XXX5
027032  005200                  INC     R0
027034  010037  002062          MOV     R0,DZMSR        ;XXX6
027040  010037  002066          MOV     R0,DZTDR        ;XXX6
027044  005200                  INC     R0
027046  010037  002064          MOV     R0,HDZMSR       ;XXX7
027052  010037  002070          MOV     R0,HDZTDR       ;XXX7
027056  000207                  RTS     PC
027060  000240  DZPRT:  PR5
027062  000200  LESS1:  PR4     ;LEVEL TO ALLOW INTERRUPTS
```

```
                                        ;ERROR ERROR TABLE
027064  000000          .ERRTAB:        0       ;ERROR 0
027066  000000                          0
027070  000000                          0

027072  027304                          EM1     ;ERROR
027074  030554                          DH1
027076  030752                          DT1

027100  027357                          EM2     ;ERROR 2
027102  030577                          DH2
027104  030764                          DT2

027106  027405                          EM3     ;ERROR 3
027110  030632                          DH3
027112  031002                          DT3

027114  027444                          EM4     ;ERROR 4
027116  030632                          DH3
027120  031002                          DT3

027122  027473                          EM5     ;ERROR 5
027124  030644                          DH4
027126  031010                          DT4

027130  027522                          EM6     ;ERROR 6
027132  030644                          DH4
027134  031010                          DT4

027136  027560                          EM7     ;ERROR 7
027140  030632                          DH3
027142  031002                          DT3

027144  027621                          EM8     ;ERROR 10
027146  030632                          DH3
027150  031002                          DT3

027152  027663                          EM9     ;ERROR 11
027154  030632                          DH3
027156  031002                          DT3

027160  027721                          EM10    ;ERROR 12
027162  030632                          DH3
027164  031002                          DT3

027166  027760                          EM13    ;ERROR 13
027170  030632                          DH3
027172  031002                          DT3

027174  030011                          EM14    ;ERROR 14
027176  030632                          DH3
027200  031002                          DT3

027202  030043                          EM15    ;ERROR 15
027204  000000                          0
027206  000000                          0
```

```
4560
4561  027210  030105                        EM16
4562  027212  030632                        DH3
4563  027214  031002                        DT3
4564
4565  027216  030156                        EM17    ;ERROR 17
4566  027220  030632                        DH3
4567  027222  031002                        DT3
4568
4569  027224  030214                        EM20
4570  027226  030632                        DH3
4571  027230  031002                        DT3
4572
4573  027232  030255                        EM21    ;ERROR 21
4574  027234  030673                        DH5
4575  027236  031026                        DT5
4576
4577  027240  030305                        EM22    ;ERROR 22
4578  027242  030644                        DH4
4579  027244  031010                        DT4
4580
4581  027246  030347                        EM23    ;ERROR 23
4582  027250  030632                        DH3
4583  027252  031002                        DT3
4584
4585  027254  030377                        EM24
4586  027256  030632                        DH3
4587  027260  031002                        DT3
4588
4589  027262  030425                        EM25
4590  027264  030632                        DH3
4591  027266  031002                        DT3
4592
4593  027270  030455                        EM26
4594  027272  030632                        DH3
4595  027274  031002                        DT3
4596
4597  027276  030504                        EM27
4598  027300  030632                        DH3
4599  027302  031002                        DT3
```

```
 4600                                        ;ERROR MESSAGES
 4601  027304 047200 020117 046123 EM1:     .ASCIZ  <200>/NO SLAVE SYNC RESPONSE FROM DZ11 REGISTER/
       027357    200 042522 044507 EM2:     .ASCIZ  <200>?REGISTER R/W FAILURE?
       027405    200 051124 047101 EM3:     .ASCIZ  <200>/TRANSMIT READY (TRDY) NOT SET/
       027444 051200 041505 044505 EM4:     .ASCIZ  <200>/RECEIVER DONE NOT SET/
       027473    200 040504 040524 EM5:     .ASCIZ  <200>/DATA COMPARISON ERROR/
       027522 042200 030532 020061 EM6:     .ASCIZ  <200>/DZ11 #RECEIVER BUFFER# ERROR/
       027560 052200 040522 051516 EM7:     .ASCIZ  <200>/TRANSMITTER FAILED TO INTERRUPT/
       027621    200 047125 054105 EM8:     .ASCIZ  <200>/UNEXPECTED TRANSMITTER INTERRUPT/
       027663    200 042522 042503 EM9:     .ASCIZ  <200>/RECEIVER FAILED TO INTERRUPT/
       027721    200 047125 054105 EM10:    .ASCIZ  <200>/UNEXPECTED RECEIVER INTERRUPT/
       027760 051600 046111 020117 EM13:    .ASCIZ  <200>/SILO ALARM SET TOO SOON/
       030011    200 044523 047514 EM14:    .ASCIZ  <200>/SILO ALARM FAILED TO SET/
       030043    200 041501 044524 EM15:    .ASCIZ  <200>/ACTION DETECTED ON INVALID LINE./
       030105    200 042522 042101 EM16:    .ASCIZ  <200>/READING DZRBUF DID NOT CLEAR SILO ALARM/
       030156 042200 052101 020101 EM17:    .ASCIZ  <200>/DATA VALID SHOULD NOT BE SET/
       030214 051200 041505 044505 EM20:    .ASCIZ  <200>/RECEIVER DONE SHOULD NOT BE SET/
       030255    200 042522 040514 EM21:    .ASCIZ  <200>/RELATIVE TIMING ERROR./
       030305    200 047515 042504 EM22:    .ASCIZ  <200>/MODEM SIGNAL ERROR ON CABLE TEST/
       030347    200 040504 040524 EM23:    .ASCIZ  <200>/DATA VALID IS NOT SET!/
       030377    200 040504 040524 EM24:    .ASCIZ  <200>/DATA OVERRUN IS SET!/
       030425    200 051106 046501 EM25:    .ASCIZ  <200>/FRAMING ERROR OCCURRED/
       030455    200 040520 044522 EM26:    .ASCIZ  <200>/PARITY ERROR OCCURRED/
       030504 043200 046125 020114 EM27:    .ASCIZ  <200>/FULL BINARY COUNT PATTERN NOT RECEIVED/

       030554 052200 040522 020120 DH1:     .ASCIZ  <200>/TRAP PC  DZ11 REG/
       030577    200 054105 042520 DH2:     .ASCIZ  <200>/EXPECTED  FOUND  REGISTER/
       030632 046200 047111 020105 DH3:     .ASCIZ  <200>/LINE NO./
       030644 042600 050130 041505 DH4:     .ASCIZ  <200>/EXPECTED  FOUND  LINE/
       030673    200 054124 046040 DH5:     .ASCIZ  <200>/TX LINE PREVIOUS TIME  ACTUAL TIME  PARAMETER/

                                   .EVEN
                                            ;DATA TABLES FOR ERROR MESSAGES
       030752 000002               DT1:     2
       030754    006     003               .BYTE   6,3
       030756 001204                       $REG1
       030760    006     001               .BYTE   6,1
       030762 001202                       $REG0

       030764 000003               DT2:     3
       030766    006     004               .BYTE   6,4
       030770 001214                       $REG5
       030772    006     001               .BYTE   6,1
       030774 001212                       $REG4
       030776    006     001               .BYTE   6,1
       031000 001202                       $REG0

       031002 000001               DT3:     1
       031004    003     001               .BYTE   3,1
       031006 001372                       SAVLIN

       031010 000003               DT4:     3
       031012    006     004               .BYTE   6,4
       031014 001214                       $REG5
       031016    006     001               .BYTE   6,1
       031020 001212                       $REG4
```

```
031022  003  001                .BYTE   3,1
031024  001372                  SAVLIN

031026  000004                 DTS:    4
031030  003  005                .BYTE   3,5
031032  001372                  SAVLIN
031034  006  011                .BYTE   6,9.
031036  001214                  $REGS
031040  006  007                .BYTE   6,7
031042  001220                  $TMP1
031044  006  001                .BYTE   6,1
031046  001400                  REGIST
                                ;TABLE OF DELAY TIMES FOR INDIVIDUAL BAUD RATES
                                ;---------------------------------------------

031050  002450                 DLYTBL: 2450                    ;TIME FOR   50 BAUD
031052  001560                         1560                    ;TIME FOR   75 BAUD
031054  001120                         1120                    ;TIME FOR  110 BAUD
031056  000750                          750                    ;TIME FOR  134 BAUD
031060  000660                          660                    ;TIME FOR  150 BAUD
031062  000330                          330                    ;TIME FOR  300 BAUD
031064  000150                          150                    ;TIME FOR  600 BAUD
031066  000060                           60                    ;TIME FOR 1200 BAUD
031070  000040                           40                    ;TIME FOR 1800 BAUD
031072  000030                           30                    ;TIME FOR 2000 BAUD
031074  000020                           20                    ;TIME FOR 2400 BAUD
031076  000010                           10                    ;TIME FOR 3600 BAUD
031100  000001                            1                    ;TIME FOR 4800 BUAD
031102  000001                            1                    ;TIME FOR 7200 BAUD
031104  000001                            1                    ;TIME FOR 9600 BAUD
031106  000001                            1                    ;TIME OF DELAY FOR 19200 BAUD

                                ;DELAYS WERE COMPUTED TO ALLOW MAXIMUM TIME AT EACH BAUD RATE
                                ;FOR ALL TESTS TO FUNCTION CORRECTLY ON A PDP11/45 WITH BIPOLAR
                                ;MEMORY. THE TIMES WERE ALSO TESTED ON AN 11/40 AND 11/10.

031110                          CORMAX:
         002362                         .=2362
002362   000240                         NOP
002364   000240                         NOP
         001512                         .=1512
001512   100000                         100000
         011572                         .=11572
011572   105720                         TSTB    (R0)+
         000001                 .END
```

```
ABASE = 160010              1#    409     450
ACDW1 = 000000            409     452
ACDW2 = 000000            409     453
ACPUOP= 000000            409     424
ACTIVE  001412            512#    786*   1834*   1835    1837*   1842    1843*   1844    1847*
ADDW0 = 000000            409     454
ADDW1 = 000000            409     455
ADDW10= 000000            409     464
ADDW11= 000000            409     465
ADDW12= 000000            409     466
ADDW13= 000000            409     467
ADDW14= 000000            409     468
ADDW15= 000000            409     469
ADDW2 = 000000            409     456
ADDW3 = 000000            409     457
ADDW4 = 000000            409     458
ADDW5 = 000000            409     459
ADDW6 = 000000            409     460
ADDW7 = 000000            409     461
ADDW8 = 000000            409     462
ADDW9 = 000000            409     463
ADEVCT= 000000            409     415
ADEVM = 000000            409     451
ADRCNT  006353           1403*   1440*   1450#
ADVANC= 104400            689#    2091    2345    2535    2549    2593    2602    2739    3279    3395    3640    3685    3799
                         3855
AENV  = 000000            409     420
AENVM = 000000            409     421
AFATAL= 000000            409     412
AMADR1= 000000            409     437
AMADR2= 000000            409     441
AMADR3= 000000            409     444
AMADR4= 000000            409     447
AMAMS1= 000000            409     431
AMAMS2= 000000            409     439
AMAMS3= 000000            409     442
AMAMS4= 000000            409     445
AMSGAD= 000000            409     417
AMSGLG= 000000            409     418
AMSGTY= 000000            409     411
AMTYP1= 000000            409     432
AMTYP2= 000000            409     440
AMTYP3= 000000            409     443
AMTYP4= 000000            409     446
APASS = 000000            409     414
APRIOR= 000000            409
APTCSU= 000040           1254    1359#
APTENV= 000001           1247    1315    1357#   1632
APTSIZ= 000200           1356#
APTSPO= 000100           1249    1317    1358#
ASWREG= 000000            409     422
ATESTN= 000000            409     413
AUNIT = 000000            409     416
AUSWR = 000000            409     423
AUTO.S  011612           1020    1940#
AVECT = 000300              1#
```

```
AVECT1= 000000   409   448
AVECT2= 000000   409   449
BAUD    024234  4080*  4400
BINWRD  006626  1526*
BIT0   = 000001   117*   194   196   198   200   237   256  2396  2551  3768  3824
BIT00  = 000001   107*   117
BIT01  = 000002   106*   116
BIT02  = 000004   105*   115
BIT03  = 000010   104*   114
BIT04  = 000020   103*   113
BIT05  = 000040   102*   112
BIT06  = 000100   101*   111
BIT07  = 000200   100*   110
BIT08  = 000400    99*   109
BIT09  = 001000    98*   108
BIT1   = 000002   116*   195   196   199   200   238   257  2396
BIT10  = 002000    97*   164   165   166   167   185   186   187   188   222   223   224   225
                   230   231   232   233   247   266   278  2396
BIT11  = 004000    96*   226   227   228   229   230   231   232   233   248   267   279  1187
                  2396  2477
BIT12  = 010000    95*   153   173   216   249   268   280  4266
BIT13  = 020000    94*   154   174   250   269   281  4170  4263
BIT14  = 040000    93*   155   175   251   270   282  1169  2033  4260
BIT15  = 100000    92*   156   176   252   271   283   887   890  1852  1983
BIT2   = 000004   115*   197   198   199   200   239   258  1073  2396
BIT3   = 000010   114*   148   203   205   207   209   240   259  3736  3738
BIT4   = 000020   113*   149   204   205   208   209   241   260  3736
BIT5   = 000040   112*   150   206   207   208   209   214   242   261  1952  1962  2033
BIT6   = 000100   111*   151   211   243   262
BIT7   = 000200   110*   152   212   244   263  1018  1662  1675  1696  1953  1960  2035
BIT8   = 000400   109*   161   163   165   167   182   184   186   188   219   221   223   225
                   227   229   231   233   245   264   276  2396
BIT9   = 001000   108*   162   163   166   167   183   184   187   188   220   221   224   225
                   228   229   232   233   246   265   277  2396
BPTVEC= 000014   124*
BRK0   = 000400   276*
BRK1   = 001000   277*
BRK2   = 002000   278*
BRK3   = 004000   279*
BRK4   = 010000   280*
BRK5   = 020000   281*
BRK6   = 040000   282*
BRK7   = 100000   283*
BRW     005232  1078   1206*
BUILD   023526  3927*  3966
BYTCNT  026176  4447*
CHRCNT  006624  1492*  1505*  1523*
CNVRT  = 104412   709*  1100   1102   1105   1108   1611   1613   1615   1670
CONVRT= 104411   707*  1030   1629
CORMAX  031110  4601*
COUNT0  023404  3794*  3850*  3884*  3892*
COUNT1  023406  3796*  3852*  3885*  3893*
CO0    = 000400   264*
CO1    = 001000   265*
CO2    = 002000   266*
CO3    = 004000   267*
```

```
C04    = 010000  268#
C05    = 020000  269#
C06    = 040000  270#
C07    = 100000  271#
CR     = 000015   32#  1293  1303
CRLF   = 000200   33#  1264  1303
CSRMAP  011620  1943#
CYCLE   011070  1079   1130  1823#
DATABP  007220  1600#  1603  1627  1630#
DATAHO  007206  1599#  1623  1626#
DCLASM= 104417   719#  2673  2760  2843  2931  2992  3061  3137  3245  3297  3427  3510  3687
                3761  3817
DCLR   = 000020   149#  1546  1547  2107  2434  2446  2454
DDISP  = 177570    39#   380
DELAY  = 104414   713#  1970  2414  2561  2567  2608  2614  2644  2694  2717  2733  2787  2810
                2858  2881  2899  2967  3016  3039  3086  3111  3172  3180  3266  3273  3327
                3343  3353  3459  3479  4132  4218
DEVADR  006350  1401#  1437  1448#
DEVICE= 104413   711#  1653  2153  2185  2217  2249  2281  2316  2362  2432  2472  2498  2537
                2632  2653  2902  3050  3122  3621  4088  4121  4142  4145  4185  4207  4227
                4251
DH1     030554  4510  4601#
DH2     030577  4514  4601#
DH3     030632  4518  4522  4534  4538  4542  4546  4550  4554  4562  4566  4570  4582  4586
                4590  4594  4598  4601#
DH4     030644  4526  4530  4578  4601#
DH5     030673  4574  4601#
DISPLA  001162   380#   802*  1199*  1642*
DISPRE  000174   342#   802
DLYCNT  006722   915*  1005*  1055  1559  1564#  3536
DLYTBL  031050  1005  4601#
DONFLG  001420   521#  2841*  2873  2895*
DSWR   = 177570    38#   379
DTR0   = 000400   245#  2341
DTR1   = 001000   246#
DTR2   = 002000   247#
DTR3   = 004000   248#
DTR4   = 010000   249#
DTR5   = 020000   250#
DTR6   = 040000   251#
DTR7   = 100000   252#
DT1     030752  4511  4601#
DT2     030764  4515  4601#
DT3     031002  4519  4523  4535  4539  4543  4547  4551  4555  4563  4567  4571  4583  4587
                4591  4595  4599  4601#
DT4     031010  4527  4531  4579  4601#
DT5     031026  4575  4601#
DVALID= 100000   176#  2477  2713  2806  2963  3370  3874
DZACTV  001404   507#   928*  1045*  1047  1823  1829  1909*  2000*  2004*  2009  2028
DZCR0   001500   569#   841   844   929  2006
DZCR1   001514   576#   930
DZCR10  001640   625#
DZCR11  001654   632#
DZCR12  001670   639#
DZCR13  001704   646#
DZCR14  001720   653#
```

```
DZCR15  001734          660*
DZCR16  001750          667*
DZCR17  001764          674*
DZCR2   001530          583*
DZCR3   001544          590*
DZCR4   001560          597*
DZCR5   001574          604*
DZCR6   001610          611*
DZCR7   001624          618*
DZCSR   002042          727*  1134  1546*  1547  1554*  2066  2106  2138  2170  2202  2234  2266  2394
                       2433  2633  2684*  2688  2692  2715  2736  2743  2777*  2781  2785  2808  2854*
                       2856  2884  2905  2943*  2965  3009*  3011  3026*  3031*  3034  3078*  3080  3092*
                       3097*  3102*  3105  3120*  3151*  3165  3170  3178  3184  3189*  3190*  3229*  3255*
                       3264  3270*  3276  3313*  3320  3325  3335  3341  3375  3401*  3405*  3410*  3445*
                       3457  3467  3473  3477  3528*  3529*  3530*  3551  3576  3584  3720*  3722  3789*
                       3792  3845*  3848  3886  3890*  3921*  3937  3979*  4061  4064  4127*  4130  4154*
                       4164  4182*  4190  4196*  4231*  4254  4289  4481*
DZLEV   026652          1856  4462*
DZLPR   002052           731*  2406*  2475*  2483*  2678*  2765*  2847*  2937*  2997*  3066*  3142*  3250*  3302*
                       3432*  3516*  3689*  3700*  3771*  3827*  4066*  4126*  4150*  4238*  4486*
DZLV0   001504          571*   866   869   887*  890*
DZLV1   001520          578*
DZLV10  001644          627*
DZLV11  001660          634*
DZLV12  001674          641*
DZLV13  001710          648*
DZLV14  001724          655*
DZLV15  001740          662*
DZLV16  001754          669*
DZLV17  001770          676*
DZLV2   001534          585*
DZLV3   001550          592*
DZLV4   001564          599*
DZLV5   001600          606*
DZLV6   001614          613*
DZLV7   001630          620*
DZMSR   002062          735*  2084  2499  2536  2596  4073*  4219  4495*
DZNUM   001410          509*   783   920*  1111  1948*  1989*  1990  1996  1998
DZPRT   027060         1850*  1851  1852*  3002*  3007  3008  3029  3030  3076  3077  3100  3101  3148
                       3150  3717  3718  3787  3788  3843  3844  4462*  4463*  4464*  4465*  4466*  4467
                       4501*
DZRBUF  002046          729*  2072  2473  2721  2775  2814  2888  2909  2944  2971  3162  3200  3371
                       3385  3450  3579  3859  3870  3940  3945  4064*  4065*  4066  4067  4167  4257
                       4485*
DZRIS   002074          741*  3007*  3029*  3076*  3100*  3148*  3230  3231*  3444*  3525*  3636  3637*  3717*
                       3787*  3843*  4152*  4233*  4472*
DZRIV   002072          740*  1137  1849*  3006*  3028*  3075*  3099*  3147*  3230*  3443*  3472*  3524*  3636*
                       3716*  3786*  3842*  4053  4151*  4232*  4470
DZTCR   002056          733*  2078  2297  2340  2407*  2639*  2687*  2726*  2780*  2821*  2853*  2871*  2879*
                       2947*  2978*  3003*  3072*  3161*  3222*  3262*  3319*  3389*  3404*  3455*  3493*  3531*
                       3571*  3719*  3790*  3846*  3919*  4067*  4068*  4069  4071  4123*  4189*  4197*  4240*
                       4491*
DZTDR   002066          737*  2501*  2508*  2698*  2745*  2791*  2876*  2949*  3032*  3103*  3176*  3271*  3331*
                       3413*  3463*  3475*  3568*  3908*  4071*  4072*  4073  4137*  4195*  4292*  4496*
DZTIS   002100          743*  3008*  3030*  3077*  3101*  3150*  3232  3233*  3527*  3638  3639*  3718*  3788*
                       3844*  4235*  4476*
```

```
DZTIV   002076            742*   3005*   3027*   3074*   3098*   3149*   3232*   3402*   3526*   3638*   3715*   3785*   3841*
                         4234*   4474*
DZV    = ****** U         1797
DZVC0   001502             570*    854     857    2049
DZVC1   001516             577*
DZVC10  001642             626*
DZVC11  001656             633*
DZVC12  001672             640*
DZVC13  001706             647*
DZVC14  001722             654*
DZVC15  001736             661*
DZVC16  001752             668*
DZVC17  001766             675*
DZVC2   001532             584*
DZVC3   001546             591*
DZVC4   001562             598*
DZVC5   001576             605*
DZVC6   001612             612*
DZVC7   001626             619*
DZ.END  002000             681*    826    1835    1844    1946
DZ.MAP  001500             512     567*    786     823    1025    1837    1847    1895    1943    1949    2020
EIGFLG  001414             517*   1851*   2343    2533    2594
EIGHT  = 000030            205*   3671    3674
EIGHTS= 000070             209*
EMTVEC= 000030             127*
EM1     027304            4509    4601*
EM10    027721            4545    4601*
EM13    027760            4549    4601*
EM14    030011            4553    4601*
EM15    030043            4557    4601*
EM16    030105            4561    4601*
EM17    030156            4565    4601*
EM2     027357            4513    4601*
EM20    030214            4569    4601*
EM21    030255            4573    4601*
EM22    030305            4577    4601*
EM23    030347            4581    4601*
EM24    030377            4585    4601*
EM25    030425            4589    4601*
EM26    030455            4593    4601*
EM27    030504            4597    4601*
EM3     027405            4517    4601*
EM4     027444            4521    4601*
EM5     027473            4525    4601*
EM6     027522            4529    4601*
EM7     027560            4533    4601*
EM8     027621            4537    4601*
EM9     027663            4541    4601*
ERRMSG  007174            1598*   1618    1621*
ERRVEC= 000004             120*   1174    1175*   1177*   1180*
ERTABO  007344            1613    1653*
EVEPAR= 000000             215*
EXITER  007300            1641    1644*
FINI    021772            3547    3635*
FIVE   = 000000             202*
FIVES  = 000040             206*
```

```
FRMERR= 020000   174*  2963  3590  3943
HALTS   007224  1583  1632*
HDRFLG  001416   519*   814*   824*   968*  1021  1023*
HDZCSR  002044   728*  2447*  2862  3554  3899  4483*
HDZLPR  002054   732*  4489*
HDZMSR  002064   736*  4498*
HDZRBU  002050   730*  4488*
HDZTCR  002060   734*  2560*  2566*  2607*  2613*  4069*  4070*  4213*  4493*
HDZTDR  002070   738*  2948*  2975*  4499*
HILIM   006346  1400*  1428  1447*
HT    = 000011    30*  1262  1303
INBUF   010620  1370  1406  1765*  1766  1771  1776  1790*  4175*  4176*  4183  4314  4366*  4367
                4371
INIFLG  001415   518*   805   810*   818   828*   954*  1016  1942*
INIT    021252  3514*  3519
INIT1   021276  3521*  3523
INSTER= 104404   697*  1422  1782  4333  4376
INSTR = 104403   695*   836   849   861   896   903   960   993  1050  1862  4048  4056  4076
                4080   890
INSTR2  006146  1377   889*
INTPAN  025456  4234  4289*
INTREC  025276  4232  4253*
INTSVC  024642  4151  4164*
IOTVEC= 000020   125*  2050*
LAST    026170  4038*  4301*  4444*
LESS1   027062  3071*  3093*  3532*  4153*  4239*  4467*  4468*  4469*  4502*
LF    = 000012    31*  1297  1303
LIMITS  006274  1416  1428*
LINE    001364   499*  1853*  2544  2598  2637  2676  2685  2763  2778  2852  2935  2945  2995
                3003  3064  3072  3140  3153  3248  3257  3300  3314  3430  3447  3511  3514
                3531  3564  3598  3625  3672  3766  3790  3822  3846
LINESP  026200  4085  4395  4448*
LINEX   024252  4088*  4160  4247
LINE0   001506   572*   912*   965   975  1001
LINE1   001522   579*
LINE10  001646   628*
LINE11  001662   635*
LINE12  001676   642*
LINE13  001712   649*
LINE14  001726   656*
LINE15  001742   663*
LINE16  001756   670*
LINE17  001772   677*
LINE2   001536   586*
LINE3   001552   593*
LINE4   001566   600*
LINE5   001602   607*
LINE6   001616   614*
LINE7   001632   621*
LOBITS  006352  1402*  1432  1449*
LOCK    001362   494*  1214  1216  1570*  1607  2065*  2071*  2077*  2083*  2299*  2328*  2342*  2381*
                2532*  2590*  2672*  2759*  2827*  2928*  2981*  3296*  3426*  3499*  3666*  4212*
LOCKUP  026166  4102*  4248*  4253*  4443*
LOLIM   006344  1399*  1430  1446*
LP0   = 000000   193*
LP1   = 000001   194*
```

```
LP2    = 000002             195#
LP3    = 000003             196#
LP4    = 000004             197#
LP5    = 000005             198#
LP6    = 000006             199#
LP7    = 000007             200#
MAINT  = 000010             148#   1204    1774    2139    2435    2448
MANT0    001512             574#    899     973    2007
MANT1    001526             581#
MANT10   001652             630#
MANT11   001666             637#
MANT12   001702             644#
MANT13   001716             651#
MANT14   001732             658#
MANT15   001746             665#
MANT16   001762             672#
MANT17   001776             679#
MANT2    001542             588#
MANT3    001556             595#
MANT4    001572             602#
MANT5    001606             609#
MANT6    001622             616#
MANT7    001636             623#
MASTEK   010315            1609    1753#
MBADLN   010424             982    1753#
MCABLE   026517            4109    4457#
MCHAR    026364            4155    4457#
MCSRX    010245            1099    1614    1753#
MDATA    010724            1503    1513    1794#
MEPASS   010063            1098    1753#
MERRPC   010372            1612    1753#
MERRX    010272            1107    1753#
MERR2    010123            1753#   1825    2011
MERR3    010172            1042    1753#
MINVAL   026310            4399    4457#
MLINE    026336            4091    4457#
MLOCK    010216            1075    1753#
MNEW     010320            1037    1753#
MNTFLG   001417             520#   1201#   1204#   1554    1769#   1774#   1779#
MODE     001370             501#   1202    1855#   2540    2591    2700    2793    2929    2951    3204    3358    3690    3759
                           3815
MPASS    026274            4300    4457#
MPASSX   010261            1104    1753#
MPFAIL   010020            1746    1753#
MQUICK   026534            4129    4457#
MR       010107            1080    1753#
MREGAD   026240            4057    4457#
MSENAB=  000040             150#   2171    2408    2409    2435    2448    2634    2640    2684    2777    2854    2943    3009
                           3031    3078    3102    3151    3255    3270    3313    3401    3410    3445    3530    3720    3789
                           3845    3979    4127    4154    4182    4196    4231
MSPEED   026346            4081    4457#
MTERM    026472            4115    4457#
MTITLE   001000             351#    809
MTSTN    010303            1610    1753#   1863
MVECTO   026216            4049    4457#
MVECX    010253            1101    1753#
```

```
MWHICH  026424          4077    4457#
NEXT    001360           493#   1569     1649    2062#   2105#   2137#   2169#   2201#   2233#   2265#   2296#   2339#   2393#
                        2431#   2471#   2497#   2531#   2542    2589#   2631#   2671#   2758#   2840#   2927#   2991#   3060#
                        3136#   3244#   3295#   3425#   3509#   3665#   3758#   3814#   4144#   4206#
NOLIST= ****** U           3
NUMLIN  026204          4124    4146    4229    4385#   4451#
NUMTCR  026206          4123    4189    4213    4215    4217    4240    4387#   4394#   4453#
ODDPAR= 000200           212#   2933    3763    3770    3772    3819    3826    3828
OFFSET  023766          3667#   3682#   3731#   3968    3970    3981#
ONESTO= 000000           213#
OTHER   024214          4040    4074#
OUT     021720          3542    3620#
OVRRUN= 040000           175#   3382    3590    3943
PAR     001366           500#   1854#   2674    2761    2844    2896    2932    2993    3062    3138    3246    3298    3428
                        3512    3762    3818    4124#   4125#   4126    4146#   4147#   4149#   4150    4228#   4229#   4230#
                        4238
PARAM = 104405           699#    838     851     863     905     962     995    1052    1864    4050    4058    4092
PARAM1  006214          1405#   1423
PARER = 010000           173#   2963    3590    3868    3874    3943
PARERR  006270          1408    1410    1412    1422#   1429    1431    1433
PARESE  023232          3786    3842    3859#
PARITY= 000100           211#   2933    3764    3820
PARMD = 104415           715#   4082
PARO    001510           573#    913#    998    1002
PAR1    001524           580#
PAR10   001650           629#
PAR11   001664           636#
PAR12   001700           643#
PAR13   001714           650#
PAR14   001730           657#
PAR15   001744           664#
PAR16   001760           671#
PAR17   001774           678#   1012
PAR2    001540           587#
PAR3    001554           594#
PAR4    001570           601#
PAR5    001604           608#
PAR6    001620           615#
PAR7    001634           622#
PAWCH = 104416           717#   4078
PIRQ  = 177772            37#
PIRQVE= 000240           131#
POPRO = 012600           140#
POP1SP= 005726           138#
POP2SP= 022626           142#    800    2046    3091    3119    3223    3228
PRIO    026210          4152    4233    4235    4454#
PRO   = 000000            54#
PR1   = 000040            55#
PR2   = 000100            56#
PR3   = 000140            57#
PR4   = 000200            58#   4502
PR5   = 000240            59#   4501
PR6   = 000300            60#
PR7   = 000340            61#    319     323     781#   1070#   2064    3158#   3444    3525    3527    3620#   4032#   4100#
                        4143#   4204#
PS    = 177776            34#     35
```

```
PSW    = 177776            35#
PUSHR0= 010046            139#
PUSH1S= 005746            137#
PUSH2S= 024646            141#
PWRVEC= 000024            126#  1715*  1716*  1725*  1731*  1743*  1744*
QUITS   025270           4251#  4283
RCVON = 010000            216#  2845   3671   3674   3668   3699   4149   4230
RDATA   026174           4446#
RDONE = 000200            152#  2396   2688   2715   2736   2781   2808   2965   3034   3105   3178   3276
RECDAT  026212           4167#  4170   4174   4175   4177*  4178*  4179   4257*  4260   4263   4266   4271   4276*
                         4277#  4278   4455#
REGIST  001400            505#  2775*  2944*  3705*  3870*  4601
RESREG  007222           1628   1631#
RESTAR  011434           1748   1888#
RESVEC= 000010            121#
RESOS = 104410            705#  1631
RIE   = 000100            151#  2235   2435   2448   3031   3102   3190   3229   3445   3528   3720   3722   3789
                         3792   3845   3848   3890   3979   4154   4196   4231
RING0 = 000001            256#
RING1 = 000002            257#
RING2 = 000004            258#
RING3 = 000010            259#
RING4 = 000020            260#
RING5 = 000040            261#
RING6 = 000100            262#
RING7 = 000200            263#
RL0   = 000000            181#
RL1   = 000400            182#
RL2   = 001000            183#
RL3   = 001400            184#
RL4   = 002000            185#
RL5   = 002400            186#
RL6   = 003000            187#
RL7   = 003400            188#
RSTART  021242           3512#
RSTRT   025512           4303#
RUN     001406            508#   787*  1829   1832*  1833*  1840*  1841*
RXISR1  023554           3716   3937#
RXSVC   021542           3524   3576#
RXTCR   022020           3511*  3540   3614*  3641#
SAVLIN  001372            502#  2635*  2654*  2682*  2699   2727*  2728   2769*  2792   2822*  2823   2842*  2886*
                         2892*  2907*  2911*  2941*  2950   2977*  3001*  3070*  3146*  3155*  3215   3254*  3259*
                         3306*  3317   3357   3391*  3436*  3453   3495*  3558*  3583*  3601   3623*  3631*  3668*
                         3701*  3702*  3776*  3832*  3863*  3901   3903*  3905*  3956*  3957*  3976*  3977*  4095
                         4179   4278   4385   4386   4601
SAVNUM  001411            510#   783*   921*  1109*  1111*  1999*
SAVPC   001402            506#  1455*  1655
SAVOS = 104407            703#  1588
SCOPI = 104401            691#  2095   2325   2368   2572   2619   2725   2746   2818   2976   3390   3411   3494
                         3730   4226
SERV.G  007360           1161   1576   1659#  1666   3729   4139   4159   4246
SET     025766           4098   4385#
SETAPT  011440            813   1895#
SETFLG= 104406            701#   898
SET.PS  010766            318   1797#
SETI    026020           4390   4393#
```

```
SEVEN = 000020            204#
SEVENS= 000060            208#
SILOAL= 020000            154#  2396   3335   3341   3375   3467   3477   3584
SILOEN= 010000            153#  2203   2435   3313   3401   3410   3445
SIX   = 000010            203#
SIXS  = 000050            207#
SNAP    021372           3535#  3615
SPACNT  006625           1524#
SPEED   026202           4125   4147   4228   4402*  4449#
SPIN    025236           4243#  4249   4286
STACK = 001120             25#   780   1069   1650   4031   4101
STFLG   026164           4034#  4089*  4106   4108*  4112   4114*  4442#
STKLMT= 177774             36#
STOP    001462            540#  2772   3309   3439   3522   3677   3708   3782   3838
SVOS    006362           1459#
SWR     001160            379#   797    801*   816    820    916    950   1018   1035   1040   1045   1048   1073
                         1169   1187   1212   1219   1577   1582   1640   1645   1647   1659   1667   1668*  1678*
                         1683*  1684*  1685*  1692*  1723   1736*  1859   1928*  4039   4074   4140
SWREG   000176            343#   801
SW0   = 000001             89#
SW00  = 000001             79#    89    816   4039
SW01  = 000002             78#    88   1859   4074
SW02  = 000004             77#    87
SW03  = 000010             76#    86    916    950
SW04  = 000020             75#    85   1048
SW05  = 000040             74#    84
SW06  = 000100             73#    83   1035
SW07  = 000200             72#    82
SW08  = 000400             71#    81   1645
SW09  = 001000             70#    80   1212
SW1   = 000002             88#
SW10  = 002000             69#  1647
SW11  = 004000             68#
SW12  = 010000             67#  1219   1577
SW13  = 020000             66#  1582
SW14  = 040000             65#
SW15  = 100000             64#
SW2   = 000004             87#
SW3   = 000010             86#
SW4   = 000020             85#
SW5   = 000040             84#
SW6   = 000100             83#
SW7   = 000200             82#
SW8   = 000400             81#
SW9   = 001000             80#
S110  = 001000            220#
S1200 = 003400            225#
S134  = 001400            221#
S150  = 002000            222#
S1800 = 004000            226#
S19200= 007400            233#
S2000 = 004400            227#
S2400 = 005000            228#
S300  = 002400            223#
S3600 = 005400            229#
S4800 = 006000            230#
```

```
S50    = 000000            218#  3671    3674
S600   = 003000            224#
S7200  = 006400            231#
S75    = 000400            219#
S9600  = 007000            232#
TABLE2   026062           4393   4408#
TBITVE= 000014              122#
TBUF     026214           4174#  4195    4456#
TCR0   = 000001            237#  2298    2407
TCR1   = 000002            238#
TCR2   = 000004            239#
TCR3   = 000010            240#
TCR4   = 000020            241#
TCR5   = 000040            242#
TCR6   = 000100            243#
TCR7   = 000200            244#
TDATA    026172           4445#
TDO      001422            524#  2770    2791    2805    2819#  3032    3103    3307    3331    3332#  3437    3463    3464#
                           3465   3475    3520    3568    3569#  3675    3706    3780    3836    3908    3909#  3911
TD1      001424            525#
TD2      001426            526#
TD3      001430            527#
TD4      001432            528#
TD5      001434            529#
TD6      001436            530#
TD7      001440            531#
TEIGHT   002140            766#
TEMP     010662           1792#
TEST1    024402           4111   4121#   4141
TEST2    025040           4105   4204#
TFIVE    002146            769#
TIE    = 040000            155#  2267    2435    3009    3026    3078    3092    3097    3189    3229    3405    3410    3529
                           3720   3789    3845    3886    3921    4231
TKVEC  = 000060            129#
TLAST  = 022772           1882   4601#
TL0    = 000000            160#
TL1    = 000400            161#
TL2    = 001000            162#
TL3    = 001400            163#
TL4    = 002000            164#
TL5    = 002400            165#
TL6    = 003000            166#
TL7    = 003400            167#
TMTBL    002102            750#  3969#   3973    3975
TPVEC  = 000064            130#
TRAPVE= 000034             128#
TRDY   = 100000            156#  2401    2405    2634    2642    2692    2743    2785    3011    3080    3325    3457    3473
TRTVEC= 000014             123#
TR0      001442            532#  3603    3605    3611#  3612    3627    3873    3876    3878    3882#  3951    3953    3959#
                           3961
TR1      001444            533#
TR2      001446            534#
TR3      001450            535#
TR4      001452            536#
TR5      001454            537#
TR6      001456            538#
```

```
TR7     001460            539#
TSEVEN  002142            767#
TSIX    002144            768#
TST1    012374           1163    1870    1886    2060#
TST10   013312           2265    2294#
TST11   013450           2296    2337#
TST12   013632           2339    2391#
TST13   013750           2393    2429#
TST14   014100           2431    2469#
TST15   014170           2471    2495#
TST16   014254           2497    2529#
TST17   014450           2531    2587#
TST2    012564           2062    2103#
TST20   014606           2589    2629#
TST21   014732           2631    2669#
TST22   015270           2671    2756#
TST23   015616           2758    2838#
TST24   016150           2840    2925#
TST25   016426           2927    2989#
TST26   016736           2991    3058#
TST27   017264           3060    3134#
TST3    012650           2105    2135#
TST30   017716           3136    3242#
TST31   020104           3244    3293#
TST32   020632           3295    3423#
TST33   021214           3425    3507#
TST34   022022           3509    3662#
TST35   022532           3665    3756#
TST36   022772           3758    3812#   4601
TST37 = ****** U         3814
TST4    012742           2137    2167#
TST5    013034           2169    2199#
TST6    013126           2201    2231#
TST7    013220           2233    2263#
TTST    005010           1076*   1078*   1165#
TWOSTO= 000040            214#   3671    3674
TXSVC   021444           3526    3551#
TYPDAT  007210           1604    1624    1627#
TYPE  = 104402            693#    809     878     981     982    1024    1037    1042    1075    1080    1098    1099    1101
                         1104    1107    1267    1368    1387    1480    1513    1605    1606    1609    1610    1612    1614
                         1616    1620    1625    1669    1671    1699    1745    1825    1861    1879    1884    2011    4109
                         4115    4155    4299    4399
TYPMSG  007100           1602    1605#
T110    002106            753#
T1200   002120            758#
T134    002110            754#
T150    002112            755#
T1800   002122            759#
T2000   002124            760#
T2400   002126            761#
T300    002114            756#
T3600   002130            762#
T4800   002132            763#
T50     002102            751#
T600    002116            757#
T7200   002134            764#
```

```
T75     002104            752#
T9600   002136            765#
VECMAP  012202           2010    2018#
VEC1    024032           4039#
VEC2    024042           4041#
WCHFLG  026162           4079    4103    4441#
WRDCNT  006622           1488*   1514*   1522#
WTBS.F  007176           1619    1622#
XBEGIN  024302           4075    4100#   4303
XBX     006766           1578    1580    1582#
XCSR    004742           1100    1132#   1615
XEOP    025476           4144    4186    4206    4252    4299#
XERR    004764           1108    1141#
XHEAD   010377           1024    1753#
XMTCNT  001376            504#   3714*   3779*   3835*   3911
XMTLIN  001374            503#   3669*   3679*   3681    3704
XMTSRV  023410           3715    3785    3841    3899#
XPASS   004756           1105    1138#
XSTART  023770            347    4031#   4033
XSTATQ  010466           1031    1753#
XTCR0   025774           4386#
XTCR1   026010           4389#   4392
XTSTN   007352           1611    1656#
XVEC    004750           1102    1135#
XX    = 160210            568#    575#    582#    589#    596#    603#    610#    617#    624#    631#    638#    645#    652#
                          659#    666#    673#    680#
YY    = 000500            568#    575#    582#    589#    596#    603#    610#    617#    624#    631#    638#    645#    652#
                          659#    666#    673#    680#
ZZ    = 000020            568#    575#    582#    589#    596#    603#    610#    617#    624#    631#    638#    645#    652#
                          659#    666#    673#    680#
$APTHD  001462            550     556#
$ASTAT= ****** U         1337    1352
$ATYC   005606           1308    1310#
$ATY1   005562           1306#
$ATY3   005570           1252    1307#
$ATY4   005600           1309#   1635
$AUTOB  001154            376#
$BASE   001310            450#    844*   1848*   1896    2006*   4480
$BDADR  001142            371#
$BDDAT  001146            373#
$CDW1   001314            452#   1920    2007*
$CDW2   001316            453#
$CHARC  005556           1269*   1279*   1286    1295*   1300#
$CMTAG  001120            359#
$CM1  = 000006            391#    392#    393#    394#    395#    396#    397#
$CM2  = 000014            391#    392#    393#    394#    395#    396#    397#
$CM3  = 000006            389#    391
$CM4  = 000004            397#    398#    399#    400#    401#
$CPUOP  001262            424#
$CRAP = 177777              1#   2054#   2058#   2097#   2101#   2127#   2133#   2159#   2165#   2191#   2197#   2223#   2229#
                         2255#   2261#   2287#   2292#   2329#   2335#   2382#   2389#   2419#   2427#   2463#   2467#   2489#
                         2493#   2515#   2526#   2575#   2585#   2622#   2627#   2658#   2667#   2749#   2754#   2830#   2836#
                         2915#   2923#   2982#   2987#   3051#   3056#   3124#   3132#   3234#   3240#   3280#   3291#   3415#
                         3421#   3500#   3505#   3644#   3660#   3743#   3754#   3800#   3810#
$CRLF   001231            403#   1268    1303    1480    1605    1606    1616    1699    1861    1879
$DDWD   001320            454#    931    1907    1944
```

```
$DDW1   001322   455#
$DDW10  001344   464#
$DDW11  001346   465#
$DDW12  001350   466#
$DDW13  001352   467#
$DDW14  001354   468#
$DDW15  001356   469#
$DDW2   001324   456#
$DDW3   001326   457#
$DDW4   001330   458#
$DDW5   001332   459#
$DDW6   001334   460#
$DDW7   001336   461#
$DDW8   001340   462#
$DDW9   001342   463#
$DEVCT  001244   415#
$DEVM   001312   451#  919#  925#  927   928   1040  1908  2009#
$DOAGN  004736  1110  1117  1122  1128#
$E    = 000040     1#  2062  2063#  2105  2106#  2137  2138#  2169  2170#  2201  2202#  2233  2234#
                2265  2266#  2296  2297#  2339  2340#  2393  2394#  2431  2432#  2471  2472#  2497
                2498#  2531  2533#  2589  2591#  2631  2632#  2671  2673#  2758  2760#  2840  2841#
                2927  2928#  2991  2992#  3060  3061#  3136  3137#  3244  3245#  3295  3296#  3425
                3426#  3509  3510#  3665  3666#  3758  3759#  3814  3815#
$ENDAD  004726   336   807  1124#  1638
$ENDCT  004712  1119#
$ENV    001254   420#  1247  1315  1339  1632
$ENVM   001255   421#  811  1249  1254  1317
$EOP    004562  1094#  3814
$EOPCT  004704  1116#  1120
$ERFLG  001123   362#  785#  1097#  1153  1183  1185#  1208  1587#  1601  1617#  4037#  4302#
$ERMAX  001135   368#  1208
$ERROR  006736   326  1576#
$ERRPC  001136   369#  789#  1096#  1162#  1584  1586#
$ERRTB  001360   488#
$ERTTL  001132   366#  788#  1143  1644#  4036#
$ETABL  001254   419#
$ETEND  001360   472#  562
$FATAL  001236   412#  1343#
$FFLG   006026  1306#  1309#  1337  1346#  1354#
$FILLC  001176   387#  1272  1303
$FILLS  001175   386#  1303
$GDADR  001140   370#
$GDDAT  001144   372#
$GET42  004716  1121#
$HD   = 000001    12  13
$HIBTS  001462   557#
$ICNT   001124   363#  1191#  1192  1194#  1207
$ILLUP  010012  1715  1731  1750#
$INTAG  001155   377#
$ITEMB  001134   367#  1592#  1634
$LF     001232   404#  1303
$LFLG   006025  1347#  1353#
$LPADR  001126   364#  791#  1079#  1081  1198#  1200  1207  1649#  1651  1877#  1878#  1886#  1888
                2542#  2543  4033#  4105#  4111#  4116
$LPERR  001130   365#
$MADR1  001266   437#
```

```
$MADR2  001272          441#
$MADR3  001276          444#
$MADR4  001302          447#
$MAIL   001234          410#    558     562     1197    1247
$MAMS1  001264          431#
$MAMS2  001270          439#
$MAMS3  001274          442#
$MAMS4  001300          445#
$MBAOR  001464          558#
$MFLG   006024         1307#   1313    1348#   1352#
$MSGAD  001250          417#   1323#   1326
$MSGLG  001252          418#   1328#
$MSGTY  001234          411#   1321    1329#   1341    1345#
$MTYP1  001265          432#
$MTYP2  001271          440#
$MTYP3  001275          443#
$MTYP4  001301          446#
$MXCNT  005234         1195    1207#   1828
$N      = 000036          1#    2054    2058    2063#   2097    2101    2106#   2127    2133    2138#   2159    2165    2170#
                       2191    2197    2202#   2223    2229    2234#   2255    2261    2266#   2287    2292    2297#   2329
                       2335    2340#   2382    2389    2394#   2419    2427    2432#   2463    2467    2472#   2489    2493
                       2498#   2516    2527    2533#   2575    2585    2591#   2622    2627    2632#   2658    2667    2673#
                       2749    2754    2760#   2830    2836    2841#   2915    2923    2928#   2982    2987    2992#   3051
                       3056    3061#   3124    3132    3137#   3234    3240    3245#   3280    3291    3296#   3415    3421
                       3426#   3500    3505    3510#   3644    3660    3666#   3743    3754    3759#   3800    3810    3815#
                       4601#
$NULL   001174          385#   1274    1303
$NWTST= 000000         2059#   2102#   2134#   2166#   2198#   2230#   2262#   2293#   2336#   2390#   2428#   2468#   2494#
                       2528#   2586#   2628#   2668#   2755#   2837#   2924#   2988#   3057#   3133#   3241#   3292#   3422#
                       3506#   3661#   3755#   3811#
$OVER   005176         1167    1170    1181    1193    1199#
$PASS   001242          414#    784#   1103#   1106#   1113#   1114#   1131    1140    1189    1208    4035#
$PASTM  001470          560#
$PWRAD  010006         1748#
$PWRDN  007646          324     782    1715#   1743
$PWRMG  010002         1746#
$PWRUP  007720         1725    1731#
$QUES   001230          402#    981    1303    1387    1884
$REGAD  001200          389#
$REG0   001202          391#   1464#   1469    4601
$REG1   001204          392#   1463#   1470    4601
$REG2   001206          393#   1462#   1471
$REG3   001210          394#   1461#   1472
$REG4   001212          395#   1460#   1473    4601
$REG5   001214          396#   1459#   1474    4601
$RTNAD  004740         1130#
$SAVR6  010016         1724#   1732    1733#   1734#   1752#
$SCOPE  004772         1160#
$SETUP= 000000         1112    1161
$SVLAD  005160         1178    1196#
$SVPC   = 000040        334#    339
$SWR    = 164000          1#     12     401     402    1091    1112    1123    1129    1131    1154    1155    1156    1157
                       1169    1181    1183    1184    1185    1186    1187    1199    1207    1749    2061    2104    2136
                       2168    2200    2232    2264    2295    2338    2392    2430    2470    2496    2530    2588    2630
                       2670    2757    2839    2926    2990    3059    3135    3243    3294    3424    3508    3663    3757
                       3813
```

```
$SWREG  001256          422*  1928
$SWRMK= 000000         1157
$TESTN  001240          413*  1197*
$TIMES  001226          401*  1112*  1186*  1192  1195*  1207  1828*  3664*
$TKB    001166          382*   882   1168  1374  1380  1661  1674  1695  4158  4245
$TKS    001164          381*   880   1166  1372  1665  1672  1693  4156  4243
$TMP0   001216          397*  3403*  3408*  3670*  3672  3683*  3719
$TMP1   001220          398*   879*   908   920   921   922*  1026*  1755  3712*  3724*  3929*  3969  3973
                       4601
$TMP2   001222          399*   927*   933*  934*  1027*  1028  1757  3927*  3930*
$TMP3   001224          400*  3713*  3726*  3928*
$TN   = 000037           12*  2059   2061*  2102  2104*  2134  2136*  2166  2168*  2198  2200*  2230  2232*
                       2262  2264*  2293  2295*  2336  2338*  2390  2392*  2428  2430*  2468  2470*  2494
                       2496*  2528  2530*  2586  2588*  2628  2630*  2668  2670*  2755  2757*  2837  2839*
                       2924  2926*  2988  2990*  3057  3059*  3133  3135*  3241  3243*  3292  3294*  3422
                       3424*  3506  3508*  3661  3663*  3755  3757*  3811  3813*
$TPB    001172          384*   886*  1292*  1303  1380*  1581*  1681*
$TPFLG  001177          388*  1241  1303
$TPS    001170          383*  1290  1303  1378  1579  1679
$TSTM   001466          559*
$TSTNM  001122          361*   790*  1153  1196*  1197  1199  1208  1658  1867  1875  2061*  2104*  2136*
                       2168*  2200*  2232*  2264*  2295*  2338*  2392*  2430*  2470*  2496*  2530*  2588*  2630*
                       2670*  2757*  2839*  2926*  2990*  3059*  3135*  3243*  3294*  3424*  3508*  3663*  3757*
                       3813*  4122*  4205*
$TYPE   005300         1241*  1334
$TYPEC  005512         1271   1278  1285  1290*  1291
$TYPEX  005560         1296   1298  1301*
$UNIT   001246          416*
$UNITM  001472          561*
$USWR   001260          423*
$VECT1  001304          448*   857*   869*  870*  871*  872*  873*  874*  1897  1900  2049*
$VECT2  001306          449*
$XTSTR  005036         1164  1172*
$Y    = 000020          682*   689   691*  693*  695*  697*  699*  701*  703*  705*  707*  709*  711*
                        713*   715*  717*  719*  721*
$$GET4= 000000         1123*
$40CAT= ****** U       1169
.     = 011574          313*   314   317*  321*  334  335*  337*  339*  341*  344*  346*  350*  358*
                        405   507*  509*  522*  546  547*  549*  551*  566*  569*  570*  571*  572*
                        573*  574*  576*  577*  578*  579*  580*  581*  583*  584*  585*  586*  587*
                        588*  590*  591*  592*  593*  594*  595*  597*  598*  599*  600*  601*  602*
                        604*  605*  606*  607*  608*  609*  611*  612*  613*  614*  615*  616*  618*
                        619*  620*  621*  622*  623*  625*  626*  627*  628*  629*  630*  632*  633*
                        634*  635*  636*  637*  639*  640*  641*  642*  643*  644*  646*  647*  648*
                        649*  650*  651*  653*  654*  655*  656*  657*  658*  660*  661*  662*  663*
                        664*  665*  667*  668*  669*  670*  671*  672*  674*  675*  676*  677*  678*
                        679*  819   1015*  1131  1207  1208  1303  1355*  1673  1680  1694  1727  1751
                       1791*  1793*  1795*  1805  1827  1969  1972  2014  2038  2379  2773  3163  3166
                       3185  3216  3220  3310  3315  3321  3336  3355  3393  3440  3448  3451  3468
                       3474  3497  3539  3552  3565  3577  3585  3588  3591  3599  3613  3628  3678
                       3709  3783  3839  4165  4168  4171  4180  4255  4258  4261  4264  4267  4274
                       4279  4290  4457*  4601*
.ADVAN  006724          690  1569*
.BEGIN  004474         1069*
.CNVRT  006452          710  1481*
.CONVR  006446          708  1480*
```

```
.DCLAS  006672             720    1553#
.DELAY  006704             714    1557#
.DEVIC  006652             712    1545#
.ERRTA  027064            1597    4505#
.INSTE  006134             698    1385#
.INSTR  006030             696    1364#
.INSTI  006050            1368#   1388
.MSG    006052            1366*   1369#
.PARAM  006154             700    1396#
.PARMD  025516             716    4306#
.PAWCH  025712             718    4365#   4377
.RESOS  006414             706    1469#
.SAVOS  006354             704    1455#
.SCOPE  004772             322    1161#   2050
.SCOP1  005236             692    1212#
.SETFL  010500             702    1764#   1783
.START  002150             345     778#    791
.TRPSR  006630             328    1534#
.TRPTA  002002             688#   1539
.TYPE   005262             694    1219#
.$ASTA= ****** U          1307    1310
.$X   = 001462             546#    551
```

```
COMMEN    132#
ENDCOM    132#
ERROR      26#  2094  2112  2125  2144  2150  2157  2176  2182  2189  2208  2214  2221  2240  2246
              2253  2272  2278  2285  2305  2313  2323  2351  2359  2367  2380  2400  2404  2417  2439
              2445  2453  2461  2480  2487  2505  2512  2565  2571  2612  2618  2647  2651  2690  2697
              2720  2724  2738  2783  2790  2813  2817  2861  2887  2893  2908  2912  2970  2974  3019
              3021  3024  3042  3044  3047  3084  3069  3095  3109  3114  3116  3164  3167  3175  3183
              3186  3194  3195  3217  3221  3226  3269  3278  3322  3330  3337  3346  3374  3377  3388
              3462  3469  3482  3487  3490  3545  3546  3553  3566  3578  3586  3589  3592  3600  3610
              3629  3728  3798  3854  3866  3875  3881  3904  3939  3942  3946  3958  3978  4135  4166
              4169  4172  4181  4194  4222  4250  4256  4259  4262  4265  4268  4275  4280  4291
ESCAPE    132#
GETPRI    132#
GETSWR    132#
MULT      132#
NEWTST    132#  2059  2102  2134  2166  2198  2230  2262  2293  2336  2390  2428  2468  2494  2528
              2586  2628  2668  2755  2837  2924  2988  3057  3133  3241  3292  3422  3506  3661  3755
              3811
PASEND      1#  1095
POP       132#  1349  1350  1736  1737
PRGEND      1#  1082
PRGFRT      1#     3
PUSH      132#  1310  1312  1333  1717  1723
REPORT      1#   132#
SC          1#  1161
SCOPE      27#  1095  2060  2103  2135  2167  2199  2231  2263  2294  2337  2391  2429  2469  2495
              2529  2587  2629  2669  2756  2838  2925  2989  3058  3134  3242  3293  3423  3507  3662
              3756  3812
SC1         1#  1201
SETPRI    132#
SETUP     132#
SKIP      132#
SLASH     132#
SPACE     132#
STARS     132#   332   354   405   408   543   545   552  1089  1150  1226  1305  1713  1729  2059
              2102  2134  2166  2198  2230  2262  2293  2336  2390  2428  2468  2494  2528  2586  2628
              2668  2755  2837  2924  2988  3057  3133  3241  3292  3422  3506  3661  3755  3811
SWRSU     132#
TYPBIN    132#
TYPDEC    132#
TYPNAM    132#
TYPNUM    132#
TYPOCS    132#
TYPOCT    132#
TYPTXT    132#
$BUFFE      1#  1787
$CYCLE      1#  1814
$EOP        1#  1082
$GETFL      1#   896
$GETPA      1#   835   848   861   903   959   993  1050  1862
$HEADE      1#    13
$INTSE      1#  3004  3027  3073  3098  3785  3841
$JUNK       1#   568   575   582   589   596   603   610   617   624   631   638   645   652   659
              666   673
$LINEU      1#  2673  2760  2992  3061  3137  3245  3297  3427
$LVLTS      1#  2982  3051
```

```
$MRESE      1#  2673  2760  2931  2992  3049  3061  3121  3137  3245  3297  3427  3510  3686  3761
            3817
$MRR        1#  2382
$MRRW       1#  2127  2159  2191  2223  2255
$MRWD       1#  2463  2489
$MSG        1#  1753
$PARTS      1#  3743  3800
$SCOPE      1#  1144
$SETFL      1#  1759
$STAG       1#  2700  2793  2951  3204  3358
$STAGF      1#
$TCR        1#  2287  2329
$TLINE      1#  2622
$TRPDE      1#   689   691   693   695   697   699   701   703   705   707   709   711   713   715
             717   719
$TSTN       1#  2058  2101  2133  2165  2197  2229  2261  2292  2335  2389  2427  2467  2493  2527
            2585  2627  2667  2754  2836  2923  2987  3056  3132  3240  3291  3421  3505  3660  3754
            3810
$UNIBU      1#  2054
$VARIA      1#   349
$XZ         1#  2054  2058  2097  2101  2127  2133  2159  2165  2191  2197  2223  2229  2255  2261
            2287  2292  2329  2335  2382  2389  2419  2427  2463  2467  2489  2493  2516  2526  2575
            2585  2622  2627  2658  2667  2749  2754  2830  2836  2915  2923  2982  2987  3051  3056
            3124  3132  3234  3240  3280  3291  3415  3421  3500  3505  3644  3660  3743  3754  3800
            3810
$$CMRE     352#   391   392   393   394   395   396
$$CMTM     352#   397   398   399   400
$$ESCA     132#
$$NEWT     132#  2059  2102  2134  2166  2198  2230  2262  2293  2336  2390  2428  2468  2494  2528
            2586  2628  2668  2755  2837  2924  2988  3057  3133  3241  3292  3422  3506  3661  3755
            3811
$$SKIP     132#
.EQUAT      1#    22
.HEADE      1#     3
.SETUP      1#
.$ACT1      1#   330
.$APTB      1#   406#
.$APTH      1#   541
.$APTY      1#  1303
.$CATC      1#
.$CMTA      1#   352
.$EOP       1#  1087
.$ERRO      1#
.$POWE      1#  1711
.$SCOP      1#  1148
.$TRAP      1#
.$TYPE      1#  1223

. ABS.   031110      000

 ERRORS DETECTED:  0

 DZDZAE,DZDZAE/SOL/CRF/NL:TOC=DZDZAE.P11
 RUN-TIME: 29 20 2 SECONDS
```

RUN-TIME RATIO: 246/52=4.6
CORE USED: 36K (71 PAGES)

EOF1DZDZAESEQ 00010000 780223 PDP10 411 1